UNIVERSIDAD AUTONOMA DE MADRID

ESCUELA POLITECNICA SUPERIOR

Grado en Ingeniería Informática

TRABAJO FIN DE GRADO

MÉTODOS DE INTELIGENCIA ARTIFICIAL APLICADOS A DATOS DE TURISMO

Autor: Alejandro Quevedo López
Tutor: David Renato Domínguez Carreta

Octubre 2019

MÉTODOS DE INTELIGENCIA ARTIFICIAL APLICADOS A DATOS DE TURISMO

AUTOR: Alejandro Quevedo López
TUTOR: David Renato Domínguez Carreta

Dpto. Ingeniería Informática
Escuela Politécnica Superior
Universidad Autónoma de Madrid
Octubre de 2019

Resumen

Durante las últimas décadas, el turismo se ha afianzado como la mayor industria mundial, habiendo experimentado un crecimiento continuado y consolidándose como uno de los pilares económicos en algunos países. El análisis de los viajes realizados y las características sociodemográficas de los turistas, así como de las regiones de destino es primordial para poder predecir tendencias e impulsar el ya afianzado desarrollo del sector.

El objetivo de este Trabado de Fin de Grado es por tanto estudiar la mayor cantidad de datos posibles obtenidos de fuentes fiables de forma que se puedan identificar algoritmos efectivos para realizar predicciones sobre el grado de satisfacción y fidelidad de los turistas hacia sus destinos elegidos.

La obtención de información sobre la que se ha trabajado ha sido limitada a España debido a la complejidad de obtener datos completos sobre viajes realizados en el resto del mundo. De esta forma se focaliza el estudio a esta región, pero dejando la posibilidad de mejorar y aumentar la cantidad de información de la base de datos muy fácilmente.

Tras la creación de esta base de datos se ha procedido a desarrollar consultas que generen los archivos sobre los que entrenarán los modelos de aprendizaje automático y así realizar el análisis utilizando la herramienta WEKA. A partir del conjunto de datos principal se han creado conjuntos de datos de entrenamiento con el número de instancias de cada clase balanceado para que los resultados obtenidos fuesen lo más fiable posible. Ha sido utilizada tanto su interfaz gráfica como la librería disponible en Java, de forma que se pudiesen automatizar al máximo los procesos a realizar.

Se han seleccionado cuatro algoritmos de clasificación de aprendizaje supervisado para cada problema: Naive Bayes, k-vecinos más próximos, árbol de decisión y perceptrón multicapa. Estos algoritmos han sido optimizados individualmente variando los valores de sus parámetros, utilizando mayoritariamente la técnica de búsqueda en cuadrícula, para determinar qué valores son los óptimos. El algoritmo que menores porcentaje de error ha devuelto para el problema de la clasificación de fidelidad es el árbol de decisión y para la clasificación de satisfacción k-vecinos más próximos. También se ha medido el rendimiento de los algoritmos con métricas como la precisión, el recall o mediante la visualización de las correspondientes matrices de confusión.

Palabras clave

Turismo, algoritmo, aprendizaje automático, perceptrón multicapa, árbol de decisión, base de datos, satisfacción, fidelidad.

Abstract

Over the last decades, tourism has become the world biggest industry, it has experimented a continuous grow and has consolidated as one of the fundamental economic pillars in some countries. The analysis of the travels tourists do and their sociodemographic characteristics, as well as those from the destination regions is capital to be able to predict tendencies and impulse the already consolidated development of the sector.

The objective of this Bachelor Thesis is then studying as much as possible data obtained from reliable sources so the most effective algorithms can be identified to make predictions of tourist satisfaction degree and fidelity to their chosen destinations.

The information used has been limited to travels in Spain due to the complexity to obtaining data from the rest of the world. This way the study is focus on this region, but letting the possibility to improve the quality of the information from the database very easily.

After the database creation, several queries have been developed to generate the files that will be used to train the machine learning modules and do the analysis using the tool WEKA. Several datasets have been created from the principal dataset with a balanced number of instances distributed in each class so the obtained results were as much reliable as possible. Both its graphic interface and its library available in Java have been used, so that the processes to perform would be automated as much as possible.

Four classification algorithms belonging to the supervised learning type have been selected for each problem: Naïve Bayes, k-Nearest Neighbors, decision tree and multilayer perceptron. These algorithms have been individually optimized varying their parameters values, mostly using the grid search technique, to determine which values are the optimums ones. The algorithm that obtained a less error for the fidelity classification problem is the decision tree y for the satisfaction classification is k-Nearest Neighbors. The performance of the algorithms has also been measured with metrics like precision, recall or visualizing the respective confusion matrices.

Keywords

Tourism, algorithm, machine learning, multilayer perceptron, decision tree, database, satisfaction, fidelity.

Agradecimientos

A mis padres por habérmelo dado todo desde el momento en el que llegué a sus vidas. A mi hermana por sus constantes muestras de cariño y palabras de ánimo. Todo lo que soy y todo lo que consiga es gracias al enorme apoyo que siempre recibo por parte de mi familia, la cual es el pilar de mi vida.

También agradecer a mis amigos más cercanos su influencia en mi mentalidad positiva a la hora de afrontar retos como este trabajo, así como sus consejos y mensajes de ánimo.

INDICE DE CONTENIDOS

INDICE DE FIGURAS

INDICE DE TABLAS

1 Introducción

1.1 Motivación

En las últimas décadas, el turismo ha crecido de forma continuada en gran parte del mundo. En algunos países este sector es incluso un factor clave de su economía y un pilar fundamental para su progreso socioeconómico. Las previsiones futuras nos indican que este crecimiento va a seguir produciéndose en los años venideros.

A día de hoy, el sector turístico se encuentra en los mismos volúmenes de negocio que el automovilístico o el petrolífero [1]. Esto nos indica su gran importancia a nivel mundial en términos de desarrollo, así como la capacidad que existe para generar ingresos de numerosos países con destinos cada vez más atractivos y asequibles para los turistas.

En España en concreto, el impacto turístico representa alrededor del 11% del producto interior bruto y se sitúa entre los tres países más visitados del mundo, siendo superado en ingresos únicamente por Estados Unidos según la OMT [2]. También sus atractivos gastronómicos y climáticos son realmente apreciados tanto por los turistas extranjeros como por los locales.

Es por todo ello que su estudio es fundamental, tanto la predicción de datos de demanda y gasto de los futuros turistas como el nivel de satisfacción y fidelidad hacia los destinos elegidos para poder obtener análisis más precisos y continuar con el fuerte desarrollo del sector dentro de este país. Este tipo de estudios serán realmente importantes, ya que ofrecerán ventajas competitivas a las empresas cuyo negocio esté basado en el turismo, provocando así un aumento sustancial de la calidad de los productos ofrecidos por el sector y en consecuencia de la satisfacción de los turistas.

Bajo estos supuestos, estudiar la mayor cantidad de datos posibles de los viajes acontecidos en España para realizar predicciones sobre el grado de satisfacción y fidelidad de los turistas en base a sus preferencias y características sociodemográficas, tanto del lugar receptor turístico como de sus clientes, se entiende como un trabajo que aporta información muy valiosa. De esta forma se podrán mejorar los servicios y productos ofrecidos con la mayor anterioridad posible, así como elaborar campañas de marketing más precisas y enfocadas en el turista, para en última instancia favorecer el ya bien encaminado desarrollo y progreso del sector.

1.2 Objetivos

El objetivo principal del TFG consiste en utilizar varias técnicas de minería de datos y algoritmos de aprendizaje automático, así como optimizarlos de forma individualizada, para evaluar su capacidad de predicción de atributos turísticos en viajes realizados por residentes en España. Para la consecución de este objetivo se estudiarán los patrones de fidelidad y grado de satisfacción de los turistas hacia los destinos escogidos en sus respectivos viajes.

Con esta información se podrá identificar en base a características e intereses qué perfiles o qué futuros turistas no acabarán completamente satisfechos una vez realizado el viaje previsto, para poder actuar en consecuencia y mejorar la experiencia y la calidad de los servicios ofrecidos, de manera que el sector siga creciendo y desarrollándose de la mejor forma posible.

Este Trabajo de Fin de Grado se ha dividido en varias partes para llevar a cabo el objetivo principal:

- La primera se trata de la obtención de información y la creación de una base de datos. Se requería gran cantidad de instancias recogidas de fuentes fiables, por lo que a pesar de que en un primer momento se pensó en recoger datos sobre viajes de todo el mundo, se redujo el alcance a España ya que la información que podía obtenerse era mucho más fidedigna y completa. De esta forma los resultados obtenidos serían más precisos. Se ha trabajado sobre ficheros de datos con más de 200.000 instancias sobre los que, una vez incorporados en bruto a una base de datos propia, se realizaron filtros específicos para eliminar atributos e instancias que no aporten valor al problema.

- La segunda consiste en el estudio de la información registrada para obtener resultados sobre la capacidad de clasificación de los diferentes algoritmos para cada uno de los patrones definidos. Se realizará un proceso de manejo de datos que incluya la creación de varios ficheros con la información normalizada y preparada para poder analizarla con la herramienta Weka. De esta forma, para cada uno de los algoritmos a estudiar se obtendrán resultados que permitirán sacar conclusiones sobre cuál es el más efectivo a la hora de clasificar a los turistas en cada una de las clases, teniendo en cuenta diferentes métricas.

1.3 Tecnología y herramientas utilizadas

Para la elaboración de este TFG se han utilizado las siguientes tecnologías y herramientas:

- *Excel*: Para el manejo de datos cuyo preprocesado era más sencillo que en la base de datos, así como la creación de gráficas y tablas con los resultados para su correcta visualización

- *PostgreSQL*: Herramienta para la creación y manejo de la base de datos con los viajes realizados en los últimos tres años en España. Uso de consultas en SQL para exportar e importar los datos e información utilizada.

- *Weka:* Herramienta gráfica para realizar pruebas con modelos de inteligencia artificial y visionado de los resultados, así como preprocesado de datos y aplicación de filtros.

- *Java y Eclipse:* Los programas para la creación de modelos de aprendizaje automático han sido creados en Java ya que la librería de Weka se encuentra en este lenguaje. La parametrización y obtención de resultados también se ha realizado en Java, con la ayuda del IDE Eclipse.

1.4 Organización de la memoria

La memoria consta de los siguientes capítulos:

- *Introducción*: Se detalla la motivación del TFG, los objetivos y las tecnologías y herramientas utilizadas durante su desarrollo.
- *Estado del arte:* Dividido en varias partes para introducir los dos aspectos principales del trabajo: turismo e inteligencia artificial. Se detalla una breve historia del turismo, sus efectos principales y el turismo en España. Se definen los algoritmos de aprendizaje automático utilizados. Finalmente se comentan las problemáticas y retos futuros.
- *Diseño y Desarrollo*: Se comenta el proceso de obtención, normalización, carga y extracción de los datos, así como su uso dentro de la base de datos y su posterior aplicación para determinar las clases a predecir. También cómo ha sido llevado a cabo el desarrollo del proyecto.
- *Integración y resultados:* Se comentan los resultados obtenidos al analizar la clasificación de las clases seleccionadas y sus atributos.
- *Conclusiones y trabajo futuro:* Se explican las conclusiones obtenidas tras la realización de este trabajo, así como las posibles ampliaciones del mismo.

2 Estado del arte

2.1 El turismo

2.1.1 Introducción

El turismo es el sector de la economía de mayor crecimiento en el mundo actual. Según la Organización Mundial del Turismo el turismo comprende "las actividades que realizan las personas durante sus viajes y estancias en lugares distintos a su entorno habitual durante un período de tiempo inferior a un año, con fines de ocio, negocios u otros". La globalización y la evolución de nuestra forma de relacionarnos han propulsado este sector hasta el punto de que en la actualidad hay países cuya economía está basada en él.

Las previsiones y los resultados obtenidos en las últimas décadas apuntan a que el desarrollo mundial del turismo va a continuar creciente, también en España [3]. Por ello entender sus causas y estudiarlo es fundamental para poder realizar análisis precisos sobre la situación económica y social del planeta.

2.1.2 Historia del turismo

El turismo como lo conocemos aparece como consecuencia de la Revolución industrial, a finales del siglo XIX. Este acontecimiento propicia la consolidación de la burguesía, de forma que ésta comienza a tener tiempo libre para realizar viajes. La intención de estos desplazamientos es el ocio, la búsqueda cultural, negocios, entre otros. Esto es debido a la evolución de los intereses y necesidades de los seres humanos, así como la capacidad de desplazarse a lugares cada vez más lejanos con mayor facilidad y rapidez.

La expansión económica acontecida en aquella época, acompañada de los cambios sociales y tecnológicos, propiciaron la aparición de este fenómeno. Por otra parte, los estallidos de las guerras mundiales repercuten fuertemente en el desarrollo del turismo, que había comenzado su expansión al inicio del siglo XX. Sin embargo, tras finalizar la Segunda Guerra Mundial, este sector comienza a crecer enormemente en todo el mundo, debido al nuevo periodo de estabilidad y orden a partir de los años 50.

Llegados los años 80, el nivel de vida occidental seguía en aumento, y el turismo se convirtió en el pilar de la economía de numerosos países como España. El sector continuaba creciendo y alcanzaba un periodo de madurez cada vez más visible, empujado por las caídas de los regímenes que quedaban en Europa.

Otros factores como la reducción de jornada laboral, las medidas sociales y conciliadoras que daban más tiempo libre a cada vez más personas, favorecieron el desarrollo del turismo hasta llegar a ser el sector más grande del mundo a finales del siglo XX y la tendencia de crecimiento sigue positiva con cifras de mejora de un 6% en el año 2018 [10].

2.1.3 Efectos del turismo

Estos efectos son tema de estudio profundo que requeriría un análisis exhaustivo que se escapa del alcance de este trabajo, por lo que simplemente se comentarán los puntos más interesantes. El turismo tiene gran impacto en los países que más lo explotan, no solo económico, sino también social, cultural y ambiental, lo cual resulta en una observación más compleja [4].

En general los efectos económicos que se producen por el turismo son bastante positivos en primera instancia. Países con grandes atractivos culturales, climáticos o gastronómicos pueden aprovechar el interés de los turistas para ingresar grandes beneficios [5]. Esto produce generación de empleo y renta, mejorando el nivel de vida de la población y la calidad de los servicios en el país. Sin embargo, también puede suponer una dependencia hacia el sector y problemas de inflación o especulación, que propician un análisis más complejo.

El impacto sociocultural del turismo en las sociedades modernas es beneficioso desde el punto de vista del turismo cultural como tal, ya que en otros tipos de turismo como el de ocio no se producen demasiados efectos al no establecerse tanta relación entre el turista y el destino. El turismo puede favorecer la conservación de espacios culturales ya sean monumentos, costumbres o tradiciones, así como ayudar a la consecución de cambios sociales y promover la diversidad, al ser punto de encuentro entre personas con mentalidades diferentes.

Por otro lado, el aumento del número de turistas y el progreso y desarrollo continuado en las últimas décadas de este sector, hacen objeto de estudio los impactos medioambientales que tiene el turismo. Un ejemplo claro de deterioro del entorno ocasionado por las visitas masivas de turistas es el de la Playa de las Catedrales, que en julio de 2015 fueron cerradas al público que no reservara con antelación [6].

Los efectos del turismo son positivos en primera instancia, pero se requiere de estudio y análisis de los mismos para controlar los impactos negativos que pueden suceder si sigue aumentando de forma continuada.

2.1.4 El turismo en España

España es el tercer país más visitado del mundo, solo por delante de Francia y Estados Unidos [7]. El agradable clima, la excelente gastronomía y el patrimonio cultural son algunos de los principales motivos por los que tanto residentes como extranjeros eligen este país para realizar turismo.

Este sector representa alrededor del 11% del PIB, habiéndose desarrollado a partir de la década de los 60, continuando con su progreso a día de hoy. El principal destino sigue siendo Cataluña, seguido por las Islas Canarias y Baleares [8]. Y según los datos del Instituto Nacional de Estadística, en 2016 el gasto de los turistas aumentó un 9%, alcanzando los 77.625 millones de euros [9].

Las previsiones indican que este fenómeno va a continuar en aumento durante los próximos años, por lo que su análisis y estudio se entiende como fundamental para favorecer la inercia positiva y poder mejorar los servicios y productos ofertados.

2.2 Aprendizaje automático

2.2.1 Introducción

El aprendizaje automático es una rama de la inteligencia artificial, la cual puede ser definida como "un campo de la ciencia y la ingeniería que se ocupa de la comprensión, desde el punto de vista informático, de lo que se denomina comúnmente comportamiento inteligente." [11]. Los programas informáticos tradicionales no mejoran su comportamiento al revisar los resultados que obtienen tras realizar una tarea. El campo de aprendizaje automático por tanto intenta sacar partido de la gran capacidad de computación y realización de tareas de los programas informáticos de forma que, a través de algoritmos e incorporación de información al sistema, mejoren su rendimiento automáticamente.

Estos tipos de sistemas se pueden clasificar dependiendo de la información que posee el algoritmo sobre el conjunto de datos. Tradicionalmente existen dos tipos diferenciados [11, p.38].

- *Aprendizaje no supervisado*: este tipo de algoritmos no conocen los datos de salida, no tienen información sobre el etiquetado de los datos de entrada. Únicamente trabajan sobre los datos de entrada buscando relaciones y estructuras entre el conjunto de datos de forma que puedan crear clases propias.

- *Aprendizaje supervisado*: el objetivo de los algoritmos de este tipo es asignar una determinada clase a nuevas instancias. Tienen acceso tanto a los valores de entrada como de salida, lo que significa que conocen de antemano la estructura de los datos. Los valores de entrada son aquella información externa que puede utilizar el algoritmo como los atributos del conjunto de datos, y los de salida son las diferentes etiquetas que puede tomar la clase. El algoritmo va aprendiendo a clasificar comparando el resultado obtenido con el valor real de la clase en cada una de las instancias, realizando correcciones del error resultante.

A través de diferentes modelos de aprendizaje automático entrenados con datos previos, se tratará obtener una clase a partir de nuevos parámetros de entrada, de forma que se pueda determinar qué modelo es el más preciso para el problema a tratar. Ya que se trata de dos problemas de clasificación para las clases de *fidelidad* y *satisfacción*, determinar si un turista va a guardar fidelidad con el destino de su viaje realizado y si va a terminar satisfecho tras realizarlo, respectivamente, el trabajo se centrará en los métodos de aprendizaje supervisado.

2.2.2 Modelos

2.2.2.1 Redes Bayesianas

Las redes bayesianas consisten en nodos y conexiones dirigidas que simbolizan las dependencias entre ellos. Cada nodo representa un atributo de interés para el problema a estudiar, como los valores de presión atmosférica en una zona para estimar la probabilidad de producirse un fenómeno como la lluvia o el viento.

La red bayesiana más sencilla es *Naive Bayes* (clasificador bayesiano ingenuo), llamado de esta forma debido a que su red asume que no hay dependencias entre atributos. En casos prácticos esto casi nunca es así, por lo tanto este algoritmo tiende a conseguir peores resultados que otros métodos más detallados.

Las redes bayesianas normales aplican automáticamente el teorema de Bayes de forma que, usando datos conocidos, estima las dependencias entre los atributos y la clase y usa esa información para calcular las probabilidades de obtener cada una de las diferentes posibles salidas de los eventos futuros:

$$\mathrm{P(A \mid B)} = \frac{P(B|A) * P(A)}{P(B)}$$

donde:

- A y B son sucesos,
- P(A) es la probabilidad de que el suceso A ocurra,
- P(A|B) es la probabilidad condicionada de que el suceso A ocurra dado que se conoce el suceso B como cierto [12].

Cada nodo del grafo se etiqueta con una distribución de probabilidad que define el efecto del nodo padre sobre el nodo hijo [13].

2.2.2.2 Aprendizaje Basado en Instancias

El aprendizaje basado en instancias consiste en el proceso de resolver problemas basándose en aquellas soluciones ya conocidas de ejercicios similares. Este tipo de modelo también se conoce como *vecino más próximo* y requiere una serie de parámetros a determinar:

- El número de vecinos que son considerados al afrontar el problema.
- Un método de evaluación que describe la función de cómo usar los vecinos encontrados para resolver el problema.
- Una función de distancia que mide la similitud entre instancias, necesaria para mediar cuáles son los vecinos más próximos en el nuevo problema.
- Una función de ponderación que habilita la cuantificación de los vecinos encontrados dependiendo de la distancia.

Este tipo de algoritmos posponen todo el trabajo hasta que se envía una consulta al sistema, por ello se encuentran dentro del grupo de los llamados *perezosos (lazy)*, en contraposición a otro tipo de algoritmos como los árboles de decisión, que intenta estructurar la información antes de recibir ninguna consulta *(eager learning methods)* [14].

2.2.2.3 Redes Neuronales Artificiales

Una red neuronal artificial se define como "un modelo matemático basado en redes neuronales biológicas, en otras palabras, es una emulación de un sistema neuronal biológico" [15, p.37]. Las redes neuronales pueden resolver tanto problemas complejos como otros más sencillos en términos de complejidad algorítmica, en contraste con otro tipo de modelos convencionales que pueden tener mayor dificultad. Es por ello por lo que el uso de redes neuronales artificiales aporta gran cantidad de valor. Su estructura simple y su capacidad para organizarse de forma automática dan la posibilidad de resolver diferentes tipos de problemas sin demasiada injerencia del programador.

Las redes neuronales pueden ser entrenadas para predecir, por ejemplo, si lloverá o no dada una serie de condiciones meteorológicas como el viento o la aparición de nubes. Estas redes consisten en nodos con conexiones entre ellos que pueden adaptar la ponderación de cada una de esas aristas durante el proceso de entrenamiento. También una función de activación que define la salida de cada nodo dependiendo de los valores que recibe en la entrada.

Estas redes tienen varias capas como se puede observar en la Figura 2-1. La capa de entrada recibe los datos de la fuente externa, la capa de salida devuelve el valor del resultado obtenido por la red y las capas ocultas conectan la capa de entrada y la capa de salida.

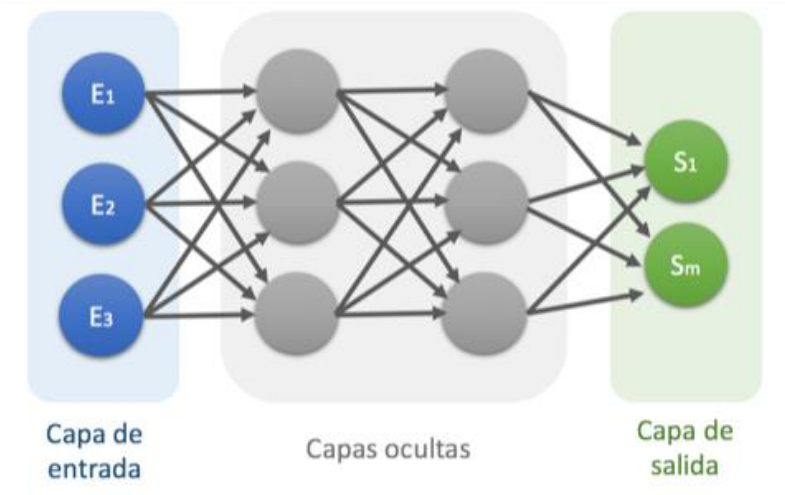

Figura 2-1 Red Neuronal Artificial

El valor de entrada de cada nodo se calcula sumando el valor de salida de cada uno de los nodos que llegan a este multiplicado por los valores de peso respectivos a la conexión entre los nodos [16, p.165]. Se pueden dividir en dos tipos [15, p.38]:

- Redes recurrentes: son todas aquellas que contienen ciclos y por tanto pueden utilizar información de etapas posteriores del proceso de aprendizaje en puntos más tempranos.

- Redes prealimentadas: estas redes no obtienen ninguna información de la propia red, por lo que los datos solo circulan en una dirección, desde los nodos de entrada hasta los nodos de salida pasando por la capa oculta con 0 hasta *n* nodos.

Para entrenar una red neuronal artificial se puede utilizar el método de propagación hacia atrás (*backpropagation).* Este proceso consiste en utilizar el error obtenido por la capa de salida propagándolo hacia atrás, de forma que llegue a todas las neuronas de la red para que, dependiendo del error relativo de cada una de ellas, los pesos que tienen sean reajustados. Esto significa que después de realizar una predicción para un conjunto de datos de entrada, el valor obtenido se compara con el valor real y se calcula el error producido. Este error se va propagando primero hacia las neuronas conectadas de forma directa con la capa de salida y posteriormente hacia las demás de la red de la misma manera [17 p.236-241].
Algunos de los parámetros importantes a definir a la hora de entrenar una red neuronal con propagación hacia atrás son los siguientes: [17 p.245]:

- Momentum: es un parámetro que sirve para agilizar el proceso de optimización de la red usando una fracción del último cambio de peso realizado y añadiéndolo al nuevo ajuste a realizar.
- Learning Rate: este parámetro indica cómo de rápido se ejecuta el proceso de aprendizaje. Varía entre 0 y 1 y es multiplicado por el error local para cada uno de los valores de salida, 0 indica que no hay adaptación y 1 una corrección total en función del error obtenido.
- Hidden Layers: el número de neuronas en las capas ocultas de la red también es importante al buscar una correcta parametrización y reducir el error resultante. Cuanto mayor sea este número más costoso será el proceso de aprendizaje al tener que pasar por un mayor número de etapas.

2.2.2.4 Árbol de Decisión

Un árbol de decisión es un modelo de predicción que consiste en fabricar diagramas de construcciones lógicas que representan y clasifican en una determinada clase el conjunto de datos proporcionado a partir de condiciones que ocurren de forma sucesiva. Se construye de forma iterativa separando el conjunto de datos sobre el atributo que mejor los divide entre las diferentes clases existentes hasta que se alcanza un determinado criterio de parada. La representación del árbol facilita al usuario observar la estructura que posee el conjunto de datos. Son árboles dirigidos que representan las reglas de decisión e ilustran las sucesivas decisiones.

En este tipo de modelos de clasificación, los nodos pueden ser identificados como raíz, interior y hoja o nodo respuesta. La raíz representa el inicio del proceso de decisión y no tiene ninguna arista entrante. Los nodos interiores tienen exactamente una arista entrante y al menos dos aristas salientes. Contienen una pregunta sobre uno de los atributos del conjunto de datos [18, p769]. Las hojas del árbol contienen la respuesta al problema de decisión, la predicción de la clase estudiada.

Por ejemplo, la figura 2-3 representa un árbol de decisión sobre un conjunto de datos con los atributos *weather (tiempo atmosférico)*, *time (tiempo)*, *hungry (hambriento)*. Una instancia con las características de *sun (soleado)* y *50 mins* pasaría por el subárbol izquierdo y obtendría un resultado de clasificación de *Walk (caminar)*.

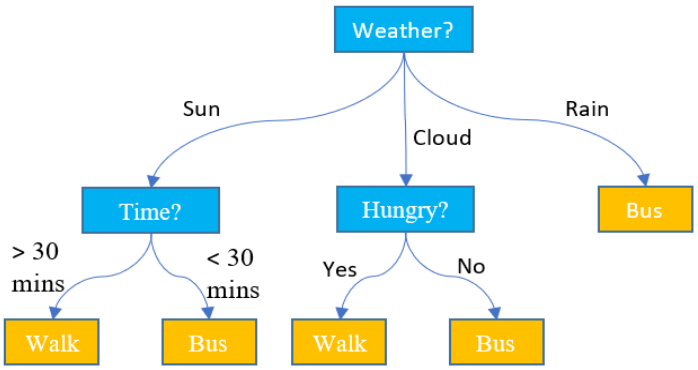

Figura 2-2 Árbol de decisión

Para entrenar un árbol de decisión y posteriormente crear un clasificador, es necesario un conjunto de datos de entrenamiento con una clase determinada, atributos y criterios de división y parada. En cada nodo, el criterio de división calcula un valor para cada uno de los atributos, que representa la cantidad de información ganada al dividir el nodo utilizando ese atributo. El mejor valor obtenido de entre todos los atributos es elegido y el nodo se divide en las respectivas salidas. Este proceso se aplica recursivamente sobre todos los subárboles generados hasta que se alcanza un criterio de parada. Un criterio de parada común es alcanzar la máxima profundidad del árbol.

El problema de entrenar los árboles de decisión de esta forma automática es que pueden generarse árboles muy grandes con secciones con muy poca capacidad de clasificación, además de llegar a estar sobreajustados, lo que significa que se adaptan demasiado a los datos de entrenamiento, resultando en clasificaciones pobres sobre nueva información. Para evitar este último problema se utiliza la técnica de *poda*, cuyo objetivo consiste en eliminar las partes menos productivas del árbol y así reducir el impacto de aquellos datos con ruido o erróneos, los cuales aparecen de forma constante en todos los conjuntos de datos [19 p.69].

2.3 Problemática y retos futuros

A pesar de que el turismo goza de previsiones realmente esperanzadoras para los próximos años, tal y como se ha ido desarrollando durante las últimas décadas, siguen existiendo problemas alrededor del sector. El análisis de los efectos negativos del turismo es fundamental para poder actuar previamente a que estos se produzcan tal y como vimos en los ejemplos anteriores. También nos encontramos con el problema de la recogida de información fiable para realizar estudios correctos en base a las características sociodemográficas de los turistas, que cada vez va solventándose más debido a las nuevas tecnologías y a los nuevos métodos de obtener datos.

Otro de los retos más importantes es la sostenibilidad y la conservación del medio ambiente y el patrimonio cultural. El aumento continuado de los turistas produce visitas cada vez más masivas a lugares que no están preparados para absorber esa cantidad de personas. Ciudades como Venecia están aplicando medidas contra este turismo masivo para intentar frenar esta degradación debido a este tipo de problemas [20].

Más allá de este tipo de problemas podemos apuntar finalmente el reto de ser capaces de mejorar los servicios y productos turísticos ofrecidos para continuar con el avance del sector y su desarrollo, de forma que la competencia suba y los turistas acaben de forma más satisfactoria sus viajes. Cada vez estos clientes son más exigentes y reclaman un mejor servicio, por ello el análisis de sus características, necesidades e intereses será de gran utilidad para empresas y organizaciones.

3 Diseño y Desarrollo

3.1 Metodología y proceso desarrollado

En este apartado se explica cómo se ha desarrollado el trabajo y llevado a cabo el proceso principal para la consecución de los objetivos indicados en la sección 1.2. Incluye la explicación del tratamiento de datos realizado así como la enumeración de los modelos de aprendizaje automático escogidos para realizar la tarea de clasificación y cómo han sido optimizados. El diseño seguido para realizar el trabajo ha sido el siguiente:

- Búsqueda y recolección de datos: la información a obtener en este punto tenía que ser de microdatos de los viajes realizados en los últimos años, que incluyese las características sociodemográficas tanto del turista como del destino. Tras realizar búsquedas en diferentes fuentes de información tanto extranjera (Eurostat, Istat, NTTO, Statistics Canada) como nacional (INE), se decidió utilizar los datos de las encuestas a residentes españoles recogidas por el Instituto Nacional de Estadística [23], con un total de 220.000 viajes realizados a lo largo de los últimos tres años (2016-2018) con 117 atributos que serán resumidos a continuación.

- Selección de las clases a analizar: una vez obtenida información suficiente, se han seleccionado dos atributos para analizar en profundidad, el nivel de satisfacción de los turistas y la fidelidad hacia el destino elegido. La asignación de las clases a predecir se ha centrado en estos dos atributos ya que son sobre los que menos información se ha encontrado y por tanto su análisis aportaría mayor valor.

- Preprocesado inicial de los datos: posteriormente se añadieron estos datos a una base de datos en PostgreSQL para poder ser tratados con mayor facilidad. Se ejecutaron consultas para filtrar la información con los datos correctos así como crear archivos .csv y tras ello transformarlos a .arff y poder realizar análisis con la librería Weka. Para crear estos archivos se han seleccionado los atributos que más información aportasen al problema.

- Elección del algoritmo de aprendizaje automático: una vez obtenidos los archivos listos para su análisis en Weka, se han seleccionado diferentes algoritmos para realizar pruebas de clasificación y determinar cuál es el óptimo para el problema a tratar.

- Interpretación de resultados: tras la obtención de los resultados devueltos por cada uno de los algoritmos en las diferentes situaciones propuestas, se han interpretado para aportar las conclusiones incluidas al final de este trabajo

3.1.1 Creación de la base de datos

La base de datos ha sido creada en PostgreSQL, utilizando los 117 atributos de los viajes que habían sido obtenidos del INE. Las tablas que actúan como diccionario para la identificación de los atributos se encuentran en el **Anexo A**. Estas son las descripciones de los más importantes:

1. **Datos sobre el destino elegido:**

    a. *CCAADEST*: Código que identifica la comunidad autónoma de destino.
    b. *PROVDEST*: Código que identifica la provincia de destino.

2. **Datos sobre las características del turista:**

    a. *EDAD*: Edad del turista.
    b. *SEXO*: Sexo del turista (1 = Hombre, 2 = Mujer).
    c. *CCAA*_RESIDENCIA: Comunidad autónoma de residencia del turista.
    d. *TAMAMU*: Tamaño del municipio de residencia. Cinco tipos de municipios dependiendo de su tamaño.
    e. *URBA*: Grado de urbanización del municipio del turista entre muy poblado, medio y escasamente poblado.
    f. *ECIVIL*: Estado civil del turista entre soltero, casado, viudo, separado, divorciado.
    g. *CONV*: Tipo de convivencia del turista entre conviviendo con su cónyuge, conviviendo con una pareja de hecho o no conviviendo en pareja.
    h. *NIVELEST*: Nivel de estudios del turista entre educación primaria, educación secundaria primera/segunda etapa y educación superior.
    i. *RELAECON*: La relación de actividad económica del turista entre ocupado, parado, jubilado o el resto de inactivos.
    j. *SITPROF*: Situación profesional del turista entre empresario o trabajador por cuenta propia que emplea/no emplea a otras personas y asalariado o trabajador por cuenta ajena con contrato indefinido/temporal.
    k. *TIPOHOGAR*: Tipo de hogar del turista entre cuatro diferentes.
    l. *VIVSEC*: La disponibilidad de vivienda secundaria (1 = Sí; 6 = No).
    m. *INGR_HOG*: Ingresos estimados totales en el hogar, divididos en 6 umbrales.

3. **Datos sobre los gastos realizados:**

    a. *GASTOFI_ALOJA*: Gasto final realizado en el alojamiento.
    b. *GASTOFI_TRANS*: Gasto final realizado en el transporte.
    c. *GASTOFI_BAREST*: Gasto final realizado en bares y restaurantes.
    d. *GASTOFI_ACT*: Gasto final realizado en actividades practicadas.
    e. *GASTOFI_TOTAL*: Gasto final total, suma del resto de gastos realizados.

4. **Datos sobre las características del viaje:**

    a. *MES*: Mes en el que se ha realizado el viaje.
    b. *ANYO*: Año de realización del viaje.
    c. *MOTIV*: Motivo principal del viaje.
    d. *TIPOVIAJ*: Clasifica los viajes atendiendo a su motivo, fechas y duración.
    e. *NPERNOC_CORR*: Pernoctaciones del viaje corregidas de outliers.
    f. *VIAJA_SOLO*: Identifica si el turista ha viajado solo (0 = No; 1 = Sí).
    g. *VIAJA_HIJOS*: Identifica si el turista ha viajado con hijos (0 = No; 1 = Sí).
    h. *ALOJAPRIN*: Tipo del alojamiento principal.
    i. *NESTR_PPAL*: Número de estrellas en el caso en el que el tipo de alojamiento principal sea hotel (de 1 a 5; 8 = No sabe).
    j. *TRANSPRIN*: Principal medio de transporte utilizado.
    k. *GRADO_SATISF*: Nivel de satisfacción del turista tras realizar el viaje entre 0 y 10 puntos.
    l. *FIDELIDAD_DEST*: Toma el valor 1 cuando el turista guarda fidelidad hacia el destino elegido y 6 cuando no.

5. **Datos sobre actividades a realizar (0 = No realiza; 1 = Realiza actividad):**

    a. *ACTI_SENDER*: Senderismo.
    b. *ACTI_AVENTURA*: Deportes de aventura/riesgo.
    c. *ACTI_VISITASCULTU*: Visitas culturales.
    d. *ACTI_CIUDADES*: Visitas a ciudades.
    e. *ACTI_FAMILIA*: Visitas a familiares y amigos.

Para ilustrar un ejemplo de turista con satisfacción sobresaliente en su viaje así como uno fiel a su destino, se representa en la siguiente tabla el valor de 7 atributos representativos. De esta forma se pretende dar una idea de cómo se podrían utilizar los resultados obtenidos en un futuro aplicativo para turistas o instituciones:

Mes	Provincia Destino	Motivo Viaje	Actividad Realizada	Sexo	Edad	CCAA Residencia	Fidelidad	Satisfacción
Julio	Málaga	Turismo deportivo	Navegación en barco	Hombre	32	Madrid	SI	Sobresaliente
Marzo	Granada	Turismo cultural	Visita cultural	Mujer	25	Sevilla	NO	Sobresaliente
Agosto	Valencia	Turismo de sol y playa	Disfrute y uso de la playa	Hombre	52	Toledo	NO	No sobresaliente

Tabla 3-1 Ejemplos fidelidad y satisfacción de turistas mostrando 7 atributos

Con este tipo de información, a través de los algoritmos de aprendizaje automático, se pueden determinar la fidelidad y el nivel de satisfacción del turista en función de sus características sociodemográficas y las del destino escogido.

3.1.2 Atribución de las clases

Para la fidelidad la atribución de la clase es directa, ya que se trata de una variable que toma valores binarios por lo que no necesitamos establecer ningún umbral. Los posibles valores son **1 = SI y 6 = NO,** es decir, un problema de clasificación binaria con los siguientes números de instancias totales en cada una de las clases:

Fidelidad	**Sí**	**No**
Valor en el dataset	1	6
Número de turistas	87980	25360
% sobre el total	77.62%	22.38%

Tabla 3-2 Distribución clase fidelidad

Se comprueba que esta distribución no está balanceada, lo cual puede producir resultados erróneos al realizar clasificaciones con los algoritmos. Para subsanar este problema se ha decidido crear de forma aleatoria conjuntos de datos utilizados para el entrenamiento de los algoritmos que tengan un número de instancias con valor de clase **Sí** igual al número de instancias con valor de clase **No**. En el apartado 4.1 de resultados se indicará el número de conjuntos de datos creados y la influencia que tiene el distribuir de esta forma los datos de entrenamiento.

El nivel de satisfacción recibe valores numéricos entre 0 y 10, por lo que la decisión sobre hacia qué tipo de clasificación aproximarse no es tan clara como para la clase anterior. En este caso, había que determinar ciertos umbrales teniendo en cuenta la cantidad de turistas que se clasificaban en cada umbral para dividirlo lo más equitativamente posible y que la distribución fuera también coherente desde el punto de vista de los sistemas de puntuación de 0 a 10:

Nivel de Satisfacción	**Medio-Bajo**	**Alto**	**Muy alto**	**Sobresaliente**
Valores umbral	<=7	8	9	10
Número de turistas	12932	33676	27607	28785
% sobre el total	12.56%	32.70%	26.80%	27.95%

Tabla 3-3 Distribución clase satisfacción (cuatro clases)

Se observa que el número de viajeros que puntúan su viaje con una valoración menor que notable es muy reducido en comparación con aquellos que sí se han sentido completamente satisfechos. Es por esto por lo que el enfoque del problema puede ser trasladado a un tipo de clasificación binaria en la que se identifique si el turista tendrá un nivel de satisfacción sobresaliente o no. La distribución de clase quedaría finalmente de la siguiente forma:

Nivel de Satisfacción	**No Sobresaliente**	**Sobresaliente**
Valores umbral	0-8	9-10
Número de turistas	46608	56392
% sobre el total	45.25%	54.75%

Tabla 3-4 Distribución clase satisfacción (dos clases)

De forma representativa, se han probado los algoritmos con la distribución de cuatro clases, con 20.000 instancias divididas en 75% para entrenamiento y 25% para test, de forma que pueda comprobarse que el error es considerablemente alto y que la aproximación al problema indicada anteriormente es más adecuada dada la distribución:

Naive Bayes				
Porcentaje de Error	F-1 Medio-Bajo	F-1 Alto	F-1 Muy Alto	F-1 Sobresaliente
72.34%	0.050	0.366	0.345	0.396

k-Vecinos Más Próximos (*k* = *2*)				
Porcentaje de Error	F-1 Medio-Bajo	F-1 Alto	F-1 Muy Alto	F-1 Sobresaliente
58.08%	0.336	0.425	0.439	0.432

Árbol de Decisión: C4.5 (*minLeafSize = 2; confidenceFactor = 0.25)*				
Porcentaje de Error	F-1 Medio-Bajo	F-1 Alto	F-1 Muy Alto	F-1 Sobresaliente
53.86%	0.332	0.491	0.474	0.501

Perceptrón Multicapa (*hiddenLayers* = 30; *learningRate* = 0.3; *momentum* = 0.2)				
Porcentaje de Error	F-1 Medio-Bajo	F-1 Alto	F-1 Muy Alto	F-1 Sobresaliente
61.19%	0.102	0.392	0.213	0.301

Tabla 3-5 Resultados clasificación satisfacción (cuatro clases)

3.1.3 Preprocesado inicial de los datos

En esta etapa lo que se ha tratado es de eliminar la información que no aportase valor al problema, tanto datos erróneos o nulos como la realización de filtros por atributos para seleccionar solo aquellos turistas cuya información del viaje realizado sea completa. Uno de los filtros realizados a destacar es por el motivo del viaje (*MOTIVO)*. Se han seleccionado solo aquellos viajes con motivo entre 0 y 9 incluidos, ya que estos son los viajes propiamente turísticos y así no se mezclaba el problema con otro tipo de datos que indiquen viajes por negocios, salud o motivos personales.

Además, han sido seleccionados únicamente 60 de los 117 atributos del conjunto de datos. Esta selección ha sido realizada de forma manual, eliminando aquellos atributos repetitivos o que no aportasen valor al problema, para así reducir el coste de la ejecución de los algoritmos. Algunos de estos atributos eliminados son por ejemplo los gastos divididos por tipos (gasto en ocio, en bares y restaurantes, etc), que en gran parte de los casos no estaban informados correctamente, ya que la suma de ellos no era igual al total indicado, por tanto únicamente se ha añadido el gasto total del viaje, el cual debería ser más preciso al tratarse de encuestas a los turistas.

Finalmente, el total de instancias sobre el que se ha realizado el análisis con los modelos de aprendizaje automático ha sido de 103.000.

3.1.4 Elección de los algoritmos de aprendizaje automático

La selección de las técnicas más apropiadas para un problema de clasificación es una parte realmente importante a la hora de llevar a cabo el proceso ya que cada una consta de ventajas y desventajas que hay que tener en cuenta. Es por ello por lo que se han elegido varios algoritmos que utilizan diferentes técnicas así como optimizaciones individualizadas para cada uno de ellos, de forma que su rendimiento pueda ser comparado y analizado:

- *Redes bayesianas*: Para esta técnica se utilizará el algoritmo Naive Bayes que, a pesar de lo explicado en el apartado 2.2.2.1 acerca de su poca precisión en los problemas de clasificación, su uso es muy común debido a los pocos recursos necesarios para poder ejecutarlo con grandes cantidades de datos. Por lo tanto, para este trabajo se ha decidido añadirlo al conjunto de algoritmos utilizados y así obtener otro enfoque en el análisis de los resultados.
  Algoritmo: Naive Bayes

- *Aprendizaje basado en instancias:* La principal ventaja del algoritmo kNN es que al ser un algoritmo perezoso no hay un proceso de aprendizaje como tal y el conjunto de datos puede aumentar de forma sencilla sin producir problemas de rendimiento. Sin embargo, al realizar la fase de los *k* vecinos más próximos cuando se lanza la consulta al sistema, el tiempo de ejecución es considerablemente alto. También se puede destacar que es sensible a distribuciones de clase no balanceadas, pero en el enfoque de este trabajo eso no se considera un problema al crear conjuntos de datos de manera aleatoria sobre el principal con las clases blanaceadas.
  Algoritmo: k vecinos más próximos (*IBk* en Weka)

- *Redes neuronales artificiales:* Este tipo de técnica tiene la ventaja de detectar relaciones no lineales entre los atributos así como las interacciones entre las variables de entrada, incorporando capas ocultas. Sin embargo, la gran desventaja es la gran cantidad de recursos consumidos lo cual causa tiempos de ejecución muy elevados y posible sobreajuste de los modelos generados [21].
  Algoritmo: Perceptrón multicapa (*MultilayerPerceptron* en Weka)

- *Árboles de decisión:* Las ventajas de este método son, como lo mencionado en el apartado 2.2.2.4, la capacidad de visualizar el modelo generado de forma que se pueda comprender y analizar la relevancia de cada uno de los atributos de forma sencilla. Por otro lado, una desventaja es el posible sobreajuste producido al entrenar los árboles de decisión, adaptándose demasiado a los datos sobre los que trabaja. Esto supone errores en la clasificación de instancias desconocidas. También la forma del árbol puede cambiar fácilmente al añadir nuevos datos y su uso de forma óptima es complejo al tener que adaptar correctamente los parámetros para generar el árbol en horquillas considerablemente amplias.
  Algoritmo: C4.5 (implementación J48 en Weka)

Para realizar las predicciones de clasificación sobre los conjuntos de datos, se ha seleccionado una implementación específica para cada uno de los algoritmos. Debido a la importancia de la correcta parametrización de los mismos se han realizado optimizaciones individualizadas siguiendo el método de búsqueda en cuadrícula o barrido de parámetros [22]. Consiste en buscar a través del conjunto de hiperparámetros del algoritmo aquella

combinación que devuelva mejores resultados al evaluar el modelo construido. Los resultados de estas pruebas están recogidos en la sección 4.1, así como las horquillas de valores de cada parámetro sobre las que se ha trabajado. Un problema producido por esta forma de realizar la optimización de parámetros es que requiere mucha capacidad de procesamiento y memoria utilizada. Algunos algoritmos como las redes neuronales requieren muchos recursos para ser construidas, por lo que el proceso de construir un modelo por cada uno de los valores posibles de cada parámetro es costoso en términos de tiempo de ejecución y de reserva de espacio de memoria y CPU.

3.1.5 Métricas para interpretación de resultados

Para poder interpretar los resultados obtenidos tras la evaluación realizada por los algoritmos sobre el conjunto de datos de test, se requieren diferentes métricas e indicadores de la calidad de las predicciones, de forma que se puedan comparar tanto las diferentes implementaciones posibles de cada algoritmo como la variación de rendimiento entre ellos. Existen multitud de factores para evaluar este problema, de entre las que destacan las siguientes:

- *Tasa de clasificación errónea*: indica la cantidad de instancias clasificadas incorrectamente sobre el total. Se calcula dividiendo el número de instancias clasificadas erróneamente por el total de instancias clasificadas en la evaluación. Esta métrica tiene un problema principal, y es que el resultado depende mayoritariamente de la distribución del número de instancias en cada clase y en el número de etiquetas posibles. Por ejemplo, si tenemos un conjunto de datos en el que el 99% de las instancias están etiquetadas en una misma clase, una tasa de clasificación errónea del 0.1 no tiene por qué ser definitiva a la hora de evaluar un algoritmo como mejor o peor que otro, ya que este error se conseguiría clasificando directamente todos los datos en aquella clase que contenga mayor número de instancias sin realizar ningún tipo de entrenamiento. Es por ello por lo que, como se explica en el apartado 3.1.2, los conjuntos de datos de entrenamiento están balanceados en cuanto a la cantidad de instancias referentes a cada una de las clases, además de la necesidad implícita del uso de más métricas.

- *Precisión*: este valor indica la cantidad de instancias clasificadas correctamente en una determinada clase sobre el total de instancias clasificadas en esa clase [24, p2]. Por ejemplo, un valor de precisión 1 para la clase fidelidad = *SI* indica que todos aquellos turistas clasificados en este tipo de fidelidad tienen realmente fidelidad hacia el destino. En el caso del ejemplo, la clase fidelidad = *NO* tendrá otro valor de precisión que no está relacionado con la precisión para la clase fidelidad = *SI*, ya que en este caso podría haber turistas clasificados con la etiqueta *NO* que sí tuviesen fidelidad y por tanto el valor de precisión para esta etiqueta sería menor a 1.

- *Recall*: este valor indica la cantidad de instancias clasificadas correctamente en una clase sobre el total de instancias clasificadas correctamente en todas las clases [24, p.2 Por ejemplo, un valor de recall 1 para la clase fidelidad = *SI* indica que todos los turistas que tienen fidelidad hacia su destino han sido clasificados en la clase *SI*. Al igual que para la precisión, este valor no tiene relación con el recall de la clase

fidelidad = *NO* ya que puede haber turistas que no tengan fidelidad etiquetados con *SI*, lo cual haría que el valor de recall de la clase fidelidad = *NO* fuese menor que 1.

- *F-Measure*: esta métrica trata de combinar precisión y recall para poder obtener un único valor que aporte información sobre la calidad de la clasificación del algoritmo calculando la media armónica entre los dos [24, p.2]. Se calcula con la siguiente formula:

$$\text{F measure} = \frac{2 * \text{precisión} * \text{recall}}{precisión + recall}$$

- *Matriz de confusión*: la matriz de confusión indica en sus columnas el número de predicciones realizada para cada clase, y en las filas el número de instancias reales de cada clase [24, p.1]. El problema de las matrices de confusion es que requieren interpretación manual, sin embargo es una de las mejores formas para observar el rendimiento de los algoritmos:

		Predicción	
Datos reales	**Positivos**	Verdaderos Positivos (VP)	Falsos Positivos (FP)
	Negativos	Falsos Negativos (FN)	Verdaderos Negativos (VN)

Tabla 3-6 Ejemplo de matriz de confusión

4 Integración y Resultados

En esta sección se explicará cómo se ha llevado a cabo la ejecución de pruebas con los algoritmos utilizados. En la parte del problema de clasificación, la cual es el punto principal del trabajo, se expondrán los resultados obtenidos y cómo se ha llegado hasta ellos: qué conjuntos de datos han sido utilizados para entrenar los algoritmos, qué parámetros han sido modificados para poder optimizarlos, cuántas optimizaciones se han realizado por cada uno de los algoritmos, qué algoritmo ha obtenido los mejores resultados en términos de rendimiento y comparación de métricas y con qué parametrización.

También se comentará los resultados obtenidos para el análisis de la serie temporal de la fidelidad de los turistas por comunidad autónoma de destino en los últimos tres años (2016-2018), mediante algoritmos de regresión, y se analizará su utilidad y fiabilidad.

4.1 Datos utilizados y proceso de pruebas

La siguiente tabla resume la cantidad de información que ha sido utilizada para el entrenamiento de los modelos de aprendizaje automático:

Número inicial de turistas del dataset principal	220.873
Número inicial de atributos del dataset principal	117
Número de datasets de entrenamiento generados	46
Número de clases a predecir	2
Número final de turistas	103.000
Número final de atributos	60

Tabla 4-1 Resumen conjunto de datos utilizado

El total de turistas inicial corresponde a la información recogida del INE sobre viajes realizados por residentes españoles entre los años 2016-2018. Este número se ha reducido a aproximadamente la mitad tras la aplicación de filtros automáticos como eliminación de duplicados, detección de valores atípicos y eliminación de atributos corrrelacionados con la clase a predecir, así como manualmente comprobando la completitud de cada uno de los atributos y el valor aportado al problema.

Finalmente el número de atributos también ha sido reducido de 117 a 70 tal y como se comentó en el apartado 3.1.3. Con ese conjunto de datos final se han seleccionado instancias de forma aleatoria para generar nuevos conjuntos de entrenamiento más reducidos para balancear el número de instancias etiquetadas en cada clase. En concreto, se han generado 10 archivos con 20.000 instancias cada uno y otros 10 con 40.000 instancias, para cada una de las clases a predecir. Para el problema de predicción de *fidelidad* se han destinado 21 conjuntos de datos (1 extra en el que se recoge la serie temporal de fidelidad por comunidad autónoma de destino) y para el de *satisfacción* los 25 restantes. Esta diferencia se debe a que la distribución de la clase *satisfacción* está prácticamente balanceada en el conjunto de datos inicial, como se comentó en el apartado 3.1.2, y daba la posibilidad a entrenar algoritmos con mayor cantidad de instancias en el conjunto de datos:

tres archivos con 60.000 instancias y dos archivos con 80.000. Esta aproximación es similar a la que se realizaría si se decide entrenar a los algoritmos mediante validación cruzada, pero ya que desde un primer momento se decidió hacerlo con un conjunto de datos de entrenamiento y otro para test (75% para entrenamiento, 25% para test), la generación de estos nuevos archivos se entendía como una aportación extra al problema, ya que permitía la posibilidad de generar más implementaciones de los algoritmos para poder comparar la eficacia de la selección de determinados hiperparámetros con mayor exhaustividad.

El proceso para evaluar los modelos de aprendizaje automático es el siguiente, habiendo utilizado la librería de Weka en Java:

1. Se carga el archivo de entrada y se asigna la clase a predecir.
2. Se normalizan los datos con valores entre 0 y 1. Los valores de los atributos son todos numéricos y el diccionario para aquellos más importantes se encuentra en el anexo A.
3. Se elige el modelo a construir.
4. Se informa del valor del parámetro elegido para optimizar el algoritmo.
5. Se entrena y se construye el modelo.
6. Se aplica el modelo y se evalúa con el 25% de los datos del archivo.
7. Se escriben los resultados (métricas mencionadas en el apartado 3.1.5).

Al utilizar la técnica de búsqueda en cuadrícula para realizar las optimizaciones de los algoritmos, los puntos 4-5-6 se realizan en bucle para cada uno de los parámetros de los algoritmos, de forma que los resultados obtenidos en cada iteración 1-7 corresponden a cada una de las variaciones del parámetro elegido sobre un mismo conjunto de datos.

Al tener diferentes posibilidades de optimización (número de parámetros modificables, horquillas de los parámetros seleccionados), cada uno de los algoritmos ha pasado un número de veces por la fase de optimización, el cual se recoge a continuación, a excepción de Naive Bayes debido a que no existe optimización posible para ese algoritmo:

	Número de modelos construidos para optimización de parámetros		
Algoritmo	**Clase Fidelidad**	**Clase Satisfacción**	**Total**
k-Vecinos Más Próximos	100	100	200
Árbol de decisión	500	500	1.000
Red Neuronal Artificial	315	63	378
Totales	915	663	**1.578**

Tabla 4-2 Resumen del número de modelos construidos para optimizar los parámetros de cada algoritmo para cada conjunto de datos de entrenamiento

De esta tabla se puede destacar que el número de optimizaciones del algoritmo kNN es menor que el del resto. Esto se debe a que para este modelo únicamente se ha optimizado un parámetro, por lo que la búsqueda en cuadrícula es menos costosa y no requiere tantas ejecuciones como en los dos casos restantes. También, hay que considerar que la optimización más costosa es la de la red neuronal, ya que la construcción de ese tipo de modelos requiere gran cantidad de recursos, como se ha comentado en el apartado 3.1.4.

4.2 Resultados de los clasificadores

4.2.1 Fidelidad al destino

La primera clase a predecir es la fidelidad de los turistas hacia el destino elegido en su viaje. A continuación se expone un resumen de los mejores resultados obtenidos ordenados por algoritmo y divididos por el número de instancias de los conjuntos de datos utilizados para después detallarlos de forma individual. También se incluye los valores de los parámetros óptimos. La tabla 4-3 contiene los porcentajes de error más bajos obtenidos así como la métrica F1 explicada en el apartado 3.1.5:

Naive Bayes			
Número de instancias	Porcentaje de Error	F-1 Fidelidad SI	F-1 Fidelidad NO
20000	22.68%	0.7645	0.7813
40000	23.20%	0.7559	0.7793
k-Vecinos Más Próximos *(k = 36)*			
Número de instancias	Porcentaje de Error	F-1 Fidelidad SI	F-1 Fidelidad NO
20000	22.28%	0.7538	0.7996
40000	22.69%	0.7426	0.7971
Árbol de Decisión: C4.5 (*minLeafSize = 86; confidenceFactor = 0.25)*			
Número de instancias	Porcentaje de Error	F-1 Fidelidad SI	F-1 Fidelidad NO
20000	21.34%	0.7658	0.8055
40000	**20.86%**	**0.7683**	0.8112
Perceptrón Multicapa (*hiddenLayers* = 11; *learningRate* = 0.5; *momentum* = 0.2)			
Número de instancias	Porcentaje de Error	F-1 Fidelidad SI	F-1 Fidelidad NO
20000	21.66%	0.767	0.7984
40000	21.16%	0.7628	**0.8121**

Tabla 4-3 Resumen de los mejores resultados obtenidos por algoritmo

En el caso del algoritmo **Naive Bayes** no existía optimización posible, por lo que se han creado los modelos utilizando los 20 conjuntos de datos generados, 10 con 20.000 instancias y otros 10 con 40.000, utilizando el 75% de la información para entrenamiento y el 25% restante para la evaluación. Sin embargo, para el resto de algoritmos sí que se han realizado optimizaciones, mediante la búsqueda en cuadrícula de los valores de parámetros que mejor rendimiento producían en cada modelo. Por lo tanto se detallan los resultados obtenidos a la hora de realizar la optimización de parámetros para cada algoritmo a continuación:

4.2.1.1 k-Vecinos Más Próximos

Para este algoritmo solo se ha optimizado uno de los parámetros, el número de vecinos. El resto de parámetros son los establecidos por defecto en Weka al seleccionar el modelo IBk. El rango que se ha utilizado es entre 1 y 100 vecinos, generando un total de 1000 optimizaciones. De los 20 conjuntos de datos destinados a realizar la optimización de los

algoritmos, se han utilizado 5 con 20.000 instancias en total, y otros 5 con 40.000 instancias. De esta forma para cada conjunto de datos se ha construido el modelo en 100 ocasiones, una por cada posible valor del parámetro utilizado. Los resultados medios obtenidos se presentan visualmente en la siguiente gráfica:

Figura 4-1 Relación entre el % de error y el número de vecinos

Se observa que a partir de k = 35 el porcentaje de error no varía mucho más, produciéndose un incremente a medida que se aumenta el número de vecinos, llegando a ser del 23.5% para 100 vecinos. Considerando también la métrica F1-measure, el número de vecinos escogido como óptimo es **36**, ya que los valores de F1 cada una de las clases son los siguientes:

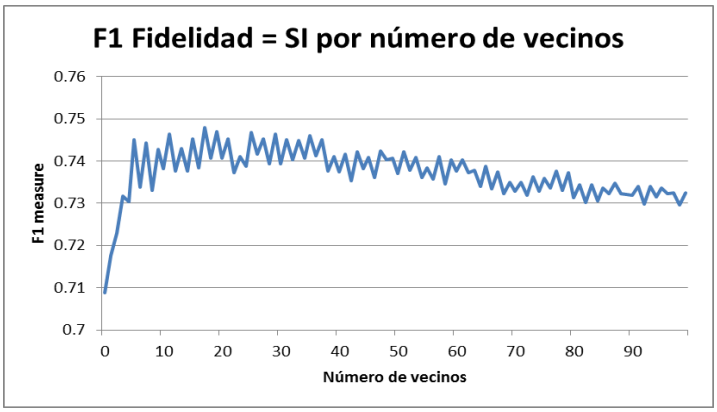

Figura 4-2 Relación entre F1-measure fidelidad = SI y el número de vecinos

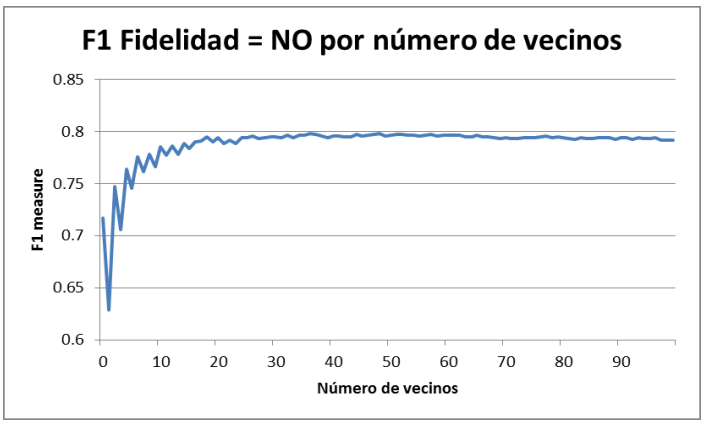

Figura 4-3 Relación entre F1-measure fidelidad = NO y el número de vecinos

Ambas medidas permanecen constantes a partir de 30 vecinos, disminuyendo una centésima la F1 para fidelidad = SI a medida que se aumenta el número de vecinos hasta llegar a 100.

4.2.1.2 Árbol de Decisión – C4.5

Para la optimización de este algoritmo se han utilizado dos parámetros: el número de observaciones por hoja (*minimal leaf size; minNumObj* en Weka) y el factor de confianza (*confidenceFactor* en Weka). El resto de parámetros son los elegidos por defecto al seleccionar la implementación del árbol de decisión J48 en Weka. El rango utilizado para el primer parámetro es entre 2 y 100, y el rango para el segundo parámetro entre 0.01 y 0.5. En total habría que generar 4.950 modelos para comprobar todas las posibles combinaciones por cada uno de los conjuntos de datos, en este caso los mismos 10 elegidos que para el algoritmo anterior, lo que haría un total de 49.500 optimizaciones. Se ha reducido este número comprobando en primera instancia que la variación del factor de confianza en el orden de centésimas no era tan relevante como se esperaba, tal y como se puede observar en la siguiente gráfica:

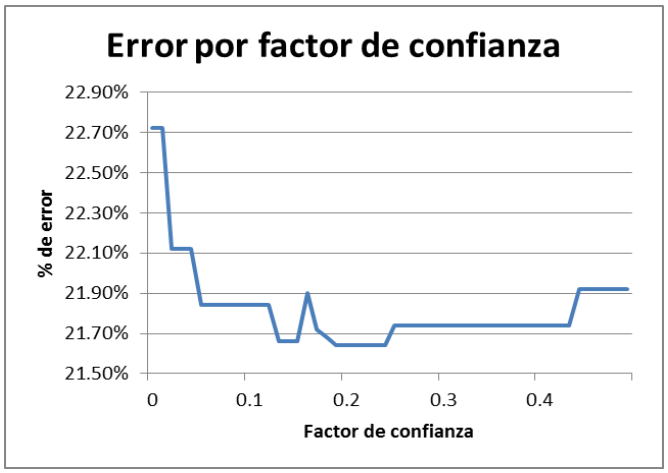

Figura 4-4 Relación entre el % de error y el valor del factor de confianza

En la figura 4-4 se comprueba que a partir de 0.1 para el factor de confianza, el error prácticamente no varía, por lo que se puede reducir el rango de optimización de este parámetro. Finalmente se ha decidido variar el factor de confianza entre 0.1 y 0.35, aumentando el valor en 0.05 puntos. Por lo tanto el total de optimizaciones realizadas para el árbol de decisión es de 5.000, construyéndose 500 modelos por cada uno de los conjuntos de datos. Como se observa en la figura 4-4, los mejores valores de porcentaje de error se centran alrededor de un valor de confianza de **0.25**, lo cual es comprobado al obtener los resultados obtenidos sobre los demás conjuntos de datos, por lo que ese valor es el elegido como óptimo.
Los resultados promedio obtenidos para la optimización del parámetro del tamaño mínimo de hoja son los siguientes:

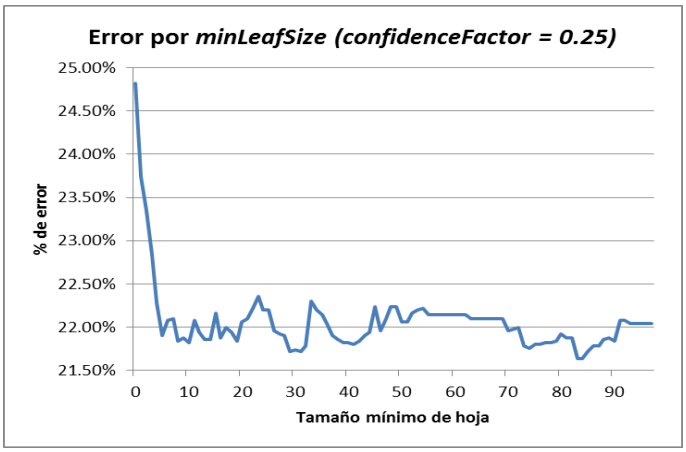

Figura 4-5 Relación entre el % de error y el tamaño mínimo de hoja

A partir del tamaño igual a 10 el porcentaje de error oscila entre el 22% de error, alcanzando el valor mínimo de 21.64% en **86**. Para asegurar que este valor es correcto utilizamos también la métrica de F1-measure, comprobando que los valores más altos para cada clase se sitúan en ese mismo punto:

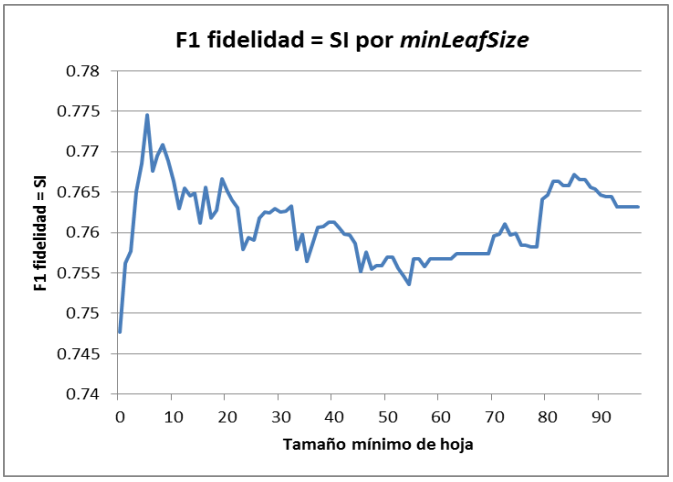

Figura 4-6 Relación entre F1-measure fidelidad = SI y el tamaño mínimo de hoja

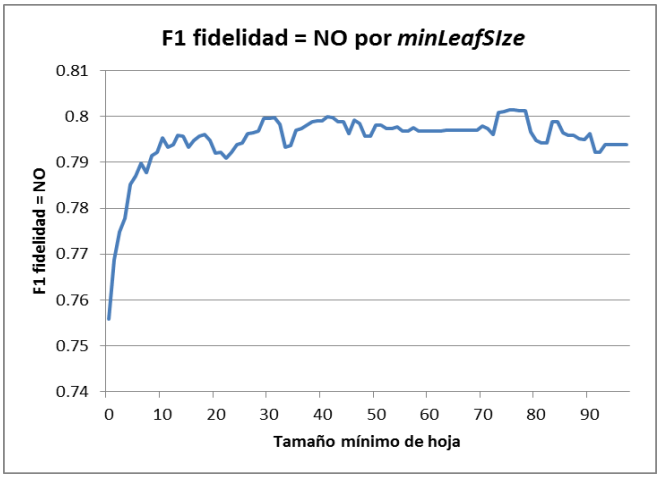

Figura 4-7 Relación entre F1-measure fidelidad = NO y el tamaño mínimo de hoja

En este caso, la medida F1-measure en la clase fidelidad = SI tiene un pico en el valor 5 del el tamaño mínimo de hoja, como se observa en la figura 4-6, sin embargo, a partir del valor 55 va incrementando de nuevo este valor hasta llegar a 87, donde consigue una F1-measure de 0.7671. Además, para la clase fidelidad = NO, la métrica F1-measure tiene un valor bajo con 5 de tamaño mínimo de hoja, por lo que se interpreta que la mejor parametrización es *minLeafSize* = **86**.

4.2.1.3 Red Neuronal Artificial – Perceptrón Multicapa

Para la optimización de este algoritmo se han utilizado tres parámetros: el número de neuronas en las capas ocultas de la red (*hiddenLayers* en Weka), la tasa de aprendizaje (*learningRate* en Weka) y el momentum. El resto de parámetros son los elegidos por defecto al seleccionar la implementación del perceptrón multicapa en Weka. El rango utilizado para el número de neuronas de la primera capa oculta es entre 1 y 100; para la tasa de aprendizaje entre 0.1 y 0.9; y para el momentum entre 0.1 y 0.7. En total habría que generar 6.300 modelos para comprobar todas las posibles combinaciones por cada uno de los conjuntos de datos, en este caso los mismos 10 elegidos que para los algoritmos anteriores, lo que haría un total de 63.000 optimizaciones. Este número es demasiado alto teniendo en cuenta lo costoso que es la construcción de una red neuronal, por lo que se ha modificado el rango de los parámetros de forma similar a lo explicado en el apartado anterior.

Primero, para establecer una horquilla menor sobre el parámetro de *hiddenLayers* se realizó una optimización individual variando entre 0 y 100 el valor para los 10 conjuntos de datos. Los resultados medios obtenidos se detallan en la siguiente gráfica (figura 4-8):

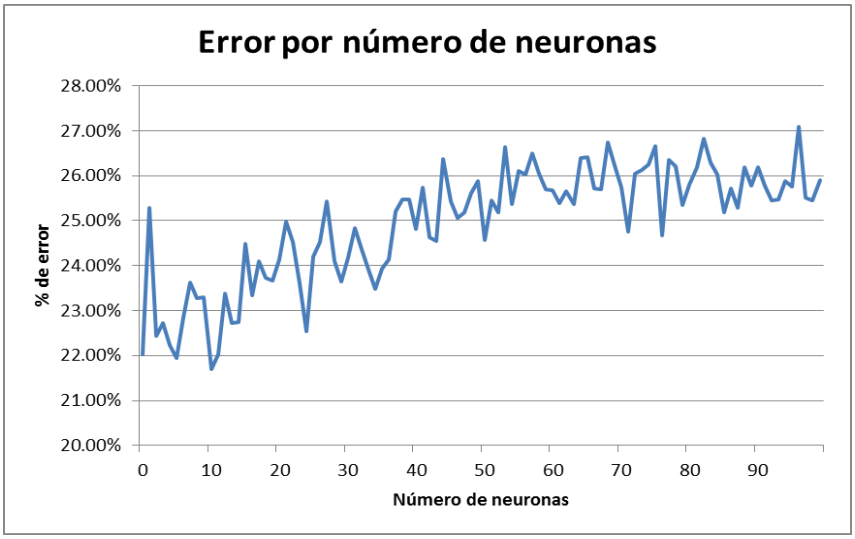

Figura 4-8 Relación entre el % de error y el número de neuronas

Se observa que los valores de menor error se centran aproximadamente en el rango de 1 a 20 neuronas, obteniendo el mejor resultado cuando *hiddenLayers* es igual a 11. Para reducir esta horquilla se utilizan 5 valores de número de neuronas para realizar la búsqueda en cuadrícula sobre la tasa de aprendizaje y momentum, de 6 a 11. Por lo tanto, el número de optimizaciones se reduce a 315 por conjunto de datos. A continuación se representa gráficamente los resultados medios de la optimización de los dos parámetros restantes:

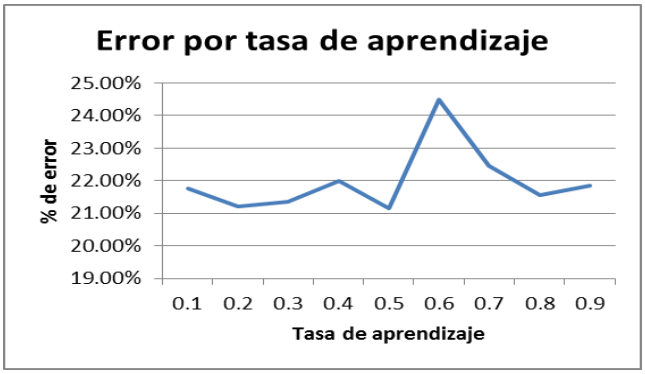

Figura 4-9 Relación entre el % de error y la tasa de aprendizaje

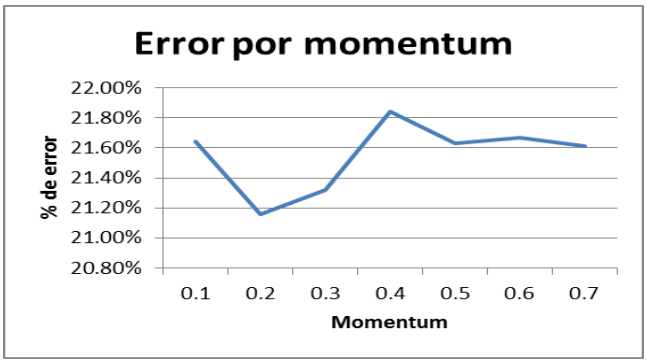

Figura 4-10 Relación entre el % de error y el momentum

Los parámetros con los que menor error se producen son por tanto *hiddenLayers* = **11**, *learningRate* = **0.5** y *momentum* = **0.2**. Para corroborar que estos parámetros son los óptimos también se utiliza la métrica F1-measure, tal y como se ha comentado en los algoritmos anteriores. Tras analizar los resultados de esta medida, se puede afirmar que los valores de los parámetros son correctos. Las gráficas con estos resultados se encuentran en el Anexo B, de forma que se puedan visualizar como para los algoritmos anteriores.

4.2.2 Nivel de satisfacción

En este apartado se presentan los resultados obtenidos para el problema de clasificar a los turistas en función de su grado de satisfacción. Al igual que en el apartado anterior, a continuación se expone un resumen de los mejores resultados obtenidos por cada uno de los algoritmos divididos por el número de instancias del conjunto de datos para después detallaros de forma individual. La tabla 4-4 contiene los porcentajes de error más bajos obtenidos así como la métrica F1 explicada en el apartado 3.1.5:

Naive Bayes			
Número de instancias	Porcentaje de Error	F-1 Sobresaliente	F-1 No Sobresaliente
20000	45.42%	0.5209	0.5957
40000	45.72%	0.5062	0.5908
60000	45.27%	0.4867	0.5788
80000	45.91%	0.4988	0.5766

k-Vecinos Más Próximos (*k = 1*)			
Número de instancias	Porcentaje de Error	F-1 Sobresaliente	F-1 No Sobresaliente
20000	37.80%	0.6193	0.6247
40000	34.26%	0.6545	0.6603
60000	32.07%	0.6768	0.6818
80000	**30.36%**	**0.6893**	**0.7031**

Árbol de Decisión: C4.5 (*minLeafSize = 5; confidenceFactor = 0.3)*			
Número de instancias	Porcentaje de Error	F-1 Sobresaliente	F-1 No Sobresaliente
20000	41.06%	0.5940	0.5890
40000	37.93%	0.6205	0.6231
60000	36.07%	0.6401	0.6384
80000	33.91%	0.6562	0.6654

Perceptrón Multicapa (*hiddenLayers* = 61; *learningRate* = 0.3; *momentum* = 0.2)			
Número de instancias	Porcentaje de Error	F-1 Sobresaliente	F-1 No Sobresaliente
20000	41.76%	0.6329	0.6201
40000	41.25%	0.5202	0.6493
60000	41.82%	0.5271	0.6262
80000	40.32%	0.4974	0.6634

Tabla 4-4 Resumen de los mejores resultados obtenidos por algoritmo

Como se comentó en el apartado 4.1, para este problema se han utilizado otros dos tipos de datasets adicionales debido a la cantidad de instancias distribuidas en cada clase. Es por ello por lo que en la tabla 4-5 se han añadido dos nuevas filas indicando los mejores resultados obtenidos para 60.000 y 80.000 instancias totales. Estos conjuntos no se han utilizado para la optimización de parámetros debido al coste en términos de recursos que suponía, pero sí para comprobar la premisa de que cuanto mayor sea el conjunto de datos de entrenamiento, mejores resultados se obtienen. De esta forma, una vez que se elegían los parámetros óptimos, se construían los modelos utilizando los conjuntos de datos restantes y se recogían los resultados, los cuales pueden observarse en la tabla 4-5.

Al igual que para el problema anterior, en el caso del algoritmo **Naive Bayes** no existía optimización posible, por lo que se han creado los modelos utilizando los 25 conjuntos de datos generados, 10 con 20.000 instancias, 10 con 40.000, 3 con 60.000 y 2 con 80.000, utilizando el 75% de la información para entrenamiento y el 25% restante para la evaluación. Sin embargo, para el resto de algoritmos sí que se han realizado optimizaciones, mediante la búsqueda en cuadrícula de los valores de parámetros que mejor rendimiento producían en cada modelo.

La metodología seguida es similar a la del problema de la clase *fidelidad*, por lo que también se incluyen gráficas de en el Anexo B. A continuación se listan los algoritmos con los resultados obtenidos de forma detallada:

4.2.2.1 k-Vecinos Más Próximos

Para este algoritmo solo se ha optimizado uno de los parámetros, el número de vecinos. El resto de parámetros son los establecidos por defecto en Weka al seleccionar el modelo IBk. El rango que se ha utilizado es entre 1 y 100 vecinos, generando un total de 1.000 optimizaciones. De los 25 conjuntos de datos destinados a realizar la optimización de los algoritmos, se han utilizado 5 con 20.000 instancias en total, y otros 5 con 40.000 instancias. De esta forma para cada conjunto de datos se ha construido el modelo en 100 ocasiones, una por cada posible valor del parámetro utilizado. Los resultados medios obtenidos se presentan visualmente en la siguiente gráfica:

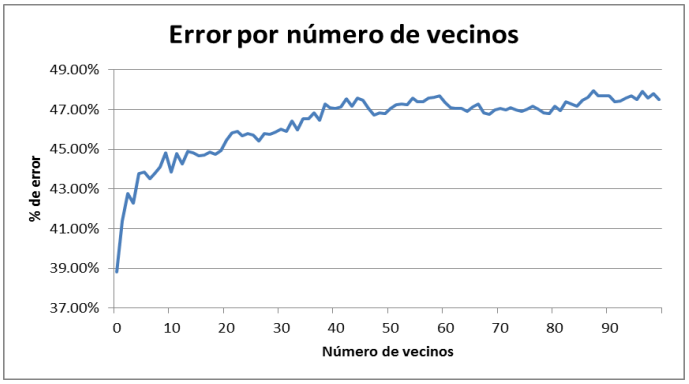

Figura 4-11 Relación entre el % de error y el número de vecinos

Se observa que a medida que el número de vecinos aumenta, también lo hace el porcentaje de error. Encontramos el menor porcentaje de error (38.82%) cuando k = 1. Para asegurar que este valor es el óptimo para el parámetro de número de vecinos consideramos también la métrica F1-measure, que alcanza sus valores máximo tanto para la clase *sobresaliente* como para *no-sobresaliente* también para k = **1**.

4.2.2.2 Árbol de Decisión – C4.5

Para la optimización de este algoritmo se han utilizado dos parámetros: el número de observaciones por hoja (*minimal leaf size; minNumObj* en Weka) y el factor de confianza (*confidenceFactor* en Weka). El resto de parámetros son los elegidos por defecto al seleccionar la implementación del árbol de decisión J48 en Weka. Como sucedía en el apartado 4.1.2.2, la variación del factor de confianza no produce demasiada alteración en el valor del porcentaje de error:

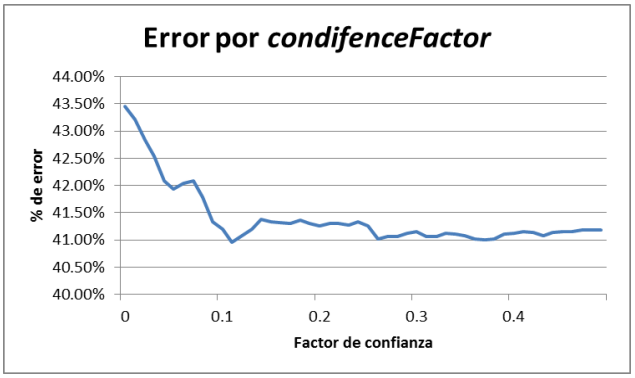

Figura 4-12 Relación entre el % de error y el factor de confianza

En la figura 4-4 se comprueba que a partir de 0.1 para el factor de confianza, el error prácticamente no varía y permanece en niveles mínimos, por lo que se en este caso también se puede reducir el rango de optimización de este parámetro a los mismos valores, entre 0.1 y 0.35, de forma que el número de optimizaciones será el mismo que para el problema de *fidelidad*. Tras realizar las optimizaciones se elige como valor óptimo para el factor de confianza **0.3**, y en la siguiente gráfica se observan los resultados promedio obtenidos para la optimización del parámetro del tamaño mínimo de hoja:

Figura 4-13 Relación entre el % de error y el tamaño mínimo de hoja

Para el valor *minLeafSize* = **5** se produce un error mínimo del 41.34%, con valores de F1-measure *sobresaliente* = 0.5874, *no-sobresaliente* = 0.5858, los cuales se encuentran entre los máximos. Por lo tanto se corrobora que es el valor óptimo para el problema a tratar.

4.2.2.3 Red Neuronal Artificial - Perceptrón Multicapa

Para la optimización de este algoritmo se han utilizado tres parámetros: el número de neuronas en las capas ocultas de la red (*hiddenLayers* en Weka), la tasa de aprendizaje (*learningRate* en Weka) y el momentum. El resto de parámetros son los elegidos por defecto al seleccionar la implementación del perceptrón multicapa en Weka. En este caso nos encontramos con el mismo problema de optimización que para la clase *fidelidad*, por lo que realizamos el mismo proceso con el primer parámetro, número de neuronas, obteniendo el siguiente resultado medio:

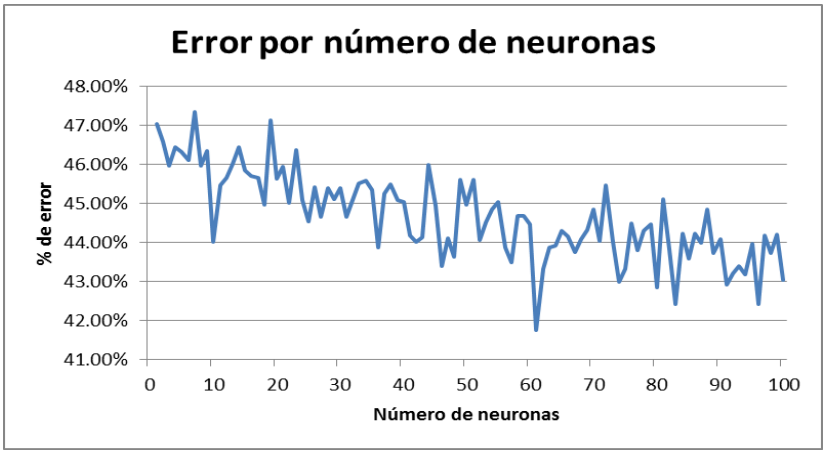

Figura 4-14 Relación entre el % de error y el número de neuronas

A medida que aumenta el número de neuronas, el porcentaje de error disminuye, siguiendo claramente una tendencia descendente. Sin embargo encontramos un pico considerable para *hiddenLayers* = **61**, por lo que tras analizar los resultados obtenidos para la métrica F1-measure se establece como valor óptimo:

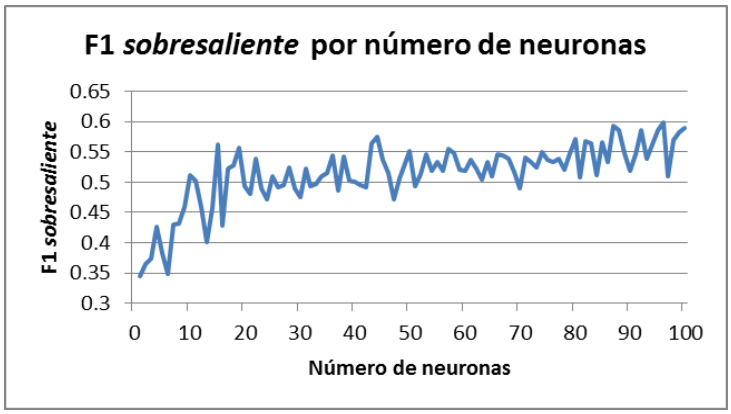

Figura 4-15 Relación entre F1-measure *sobresaliente* y número de neuronas

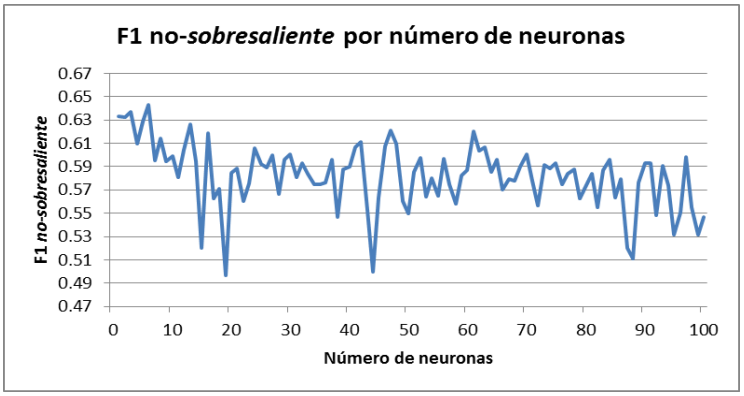

Figura 4-16 Relación entre F1-measure *sobresaliente* y número de neuronas

En la gráfica 4-15 se observa una tendencia levemente ascendente, al contrario que en la gráfica 4-16 que es descendente, por tanto el valor óptimo se encuentra alrededor de las 50 neuronas en la capa oculta. Debido al coste de procesamiento para generar las redes neuronales y al resultado de porcentaje de error obtenido, se decide utilizar el parámetro *hiddenLayers* con un valor de 61, considerándolo óptimo para el problema a tratar. De esta forma se generarán 63 optimizaciones por cada conjunto de datos estableciendo el rango de valores de *learningRate* entre 0.1 y 0.9 y de *momentum* entre 0.1 y 0.7. Por tanto, tras realizar las optimizaciones correspondientes, se determina que el valor óptimo para la tasa de aprendizaje es de **0.3**, para el cual el porcentaje de error es de 42.96%, F1-measure *sobresaliente* igual a 0.5568 y F1-measure *no-sobresaliente* igual a 0.6194. Y para el parámetro *momentum* el valor óptimo se determina como **0.2**, con un porcentaje de error de 41.76%, F1-measure *sobresaliente* igual a 0.5365 y F1-measure *no-sobresaliente* igual a 0.6201.
Las gráficas de resumen de porcentaje de error y de F1-measure para cada uno de esos dos parámetros se pueden visualizar en el Anexo B.

5 Conclusiones y trabajo futuro

5.1 Conclusiones

La elaboración de este trabajo ha supuesto un estudio de datos de viajes turísticos basados en las características sociodemográficas de los turistas, así como de los atributos de las regiones receptoras. Se ha utilizado información de los viajes realizados por turistas españoles de los últimos tres años (2016-2018), obtenida del Instituto Nacional de Estadísticas. Estos microdatos constaban de un total de 220.873 instancias y 117 atributos, que finalmente han sido reducidos a 103.000 y 60 respectivamente tras realizar el preprocesado y aplicar diferentes filtros.

Se han seleccionado las clases *fidelidad* y *satisfacción* para aplicar modelos de clasificación, por lo tanto el problema a tratar se ha centrado en los algoritmos de aprendizaje automático dedicados a la clasificación. El estudio se ha realizado con cuatro de los más conocidos, cada uno en su respectiva implementación de la librería Weka, la herramienta de aprendizaje automático utilizada en el trabajo. Cada uno de ellos ha sido optimizado variando el valor de sus parámetros realizando mayoritariamente una búsqueda en cuadrícula para encontrar los valores óptimos, a excepción de Naive Bayes que no tiene optimización posible.

El mejor resultado obtenido para la clasificación de los turistas según tengan o no fidelidad hacia el destino es de un 20.86% de error, utilizando el árbol de decisión con los parámetros minLeafSize = 86 y confidenceFactor = 0.25. Para la métrica F1-measure, los mejores resultados son 0.7683 para *fidelidad = SI* y 0.8121 para *fidelidad = NO*, obtenidos por el árbol de decisión y el perceptrón multicapa respectivamente. Con ello se destaca la importancia de utilizar diferentes métricas a la hora de valorar los clasificadores.

Por otro lado, para la clasificación de los turistas según su nivel de satisfacción sobre el viaje realizado el mejor resultado obtenido ha sido de 30.36% de error, utilizando el algoritmo k-Vecinos más próximos con k = 1. En este caso, el mejor resultado de la métrica F1-measure también es obtenido por este algoritmo, con resultados de 0.6893 para *satisfacción = sobresaliente* y de 0.7031 para *satisfacción = no-sobresaliente*. Este resultado indica que no existe prácticamente ruido en el conjunto de datos, al solo seleccionar el vecino más cercano.

Al comparar los resultados de los dos problemas de clasificación se puede destacar la importancia del uso de varios métodos de clasificación debido a la singularidad de cada problema. También comprobar que, salvo en algunas ocasiones, el aumento de la cantidad de datos utilizados para el entrenamiento de los algoritmos produce mejoras significativas en el rendimiento de los mismos, pudiéndose producir sin embargo sobreajustes.

Esta información proporciona valor tanto a los candidatos a turista de manera que puedan estimar si un viaje les es rentable en base a sus características con anterioridad, como a las empresas y organizaciones que quieran aumentar su competitividad y rentabilidad

anticipando hacia qué perfiles tienen que orientar la mejora de servicios o dar una atención especial.

La fidelidad y el grado de satisfacción de los turistas son características primordiales en el desarrollo del turismo y, a pesar de la tendencia creciente que sigue teniendo el sector, todo tipo de ayudas para mejorar e innovar en este ámbito se ven como realmente positivas, ya que son un impulso para superar los futuros retos y problemas por venir.

Cabe destacar que la parte de obtención de datos ha sido compleja, esa información fiable y suficiente sobre la que estudiar los diferentes algoritmos. La mayoría de organizaciones y gobiernos proporcionan microdatos de los viajes a través de peticiones formales que indiquen el uso que se le dará a esos datos, es por ello que se limitó el alcance a viajes realizados en España, teniendo en cuenta también la importancia del sector en este país y la posibilidad de aplicar las técnicas a valores relacionados con el agrado que le supone al turista realizar un viaje.

Para finalizar comentar que la realización de este TFG ha supuesto un fuerte refuerzo en los conocimientos previos que tenía de Java y bases de datos, así como un aprendizaje en el ámbito de la inteligencia artificial y uso de modelos predictivos que desconocía previamente.

5.2 Trabajo futuro

Existen múltiples posibilidades para ampliar y mejorar el trabajo realizado, debido a la extensión del campo y las numerosas aplicaciones que puede llegar a tener un aplicativo:

- Aumentar la cantidad de información recogida y extender el alcance a más países europeos o del mundo (espacio). De esta forma no solo se conseguiría aumentar el alcance del proyecto, sino que también se aumentaría la calidad de las predicciones al aportar muchos más datos y con mayor diversidad a los entrenamientos de los algoritmos. Este punto sería cable en una futura mejora del trabajo realizado, ya que sin necesidad de modificar implementaciones la fiabilidad sería mayor y el aplicativo mucho más útil al cubrir más espacio.

- Ampliar el alcance añadiendo el estudio de series temporales (tiempo). Así se podrá extraer una mayor cantidad de información de los atributos y clases analizadas en función de la variable temporal.

- Estudiar un mayor número de algoritmos de aprendizaje automático, así como analizar posibles mejoras de implementación de los ya utilizados. Se mejoraría la precisión de los algoritmos y se podría hacer una elección aún más estudiada y analizada de qué modelos a utilizar son los más interesantes.

- Añadir más clases sobre las que realizar predicciones. Puede ser interesante conocer de antemano otro tipo de atributos como el gasto total del turista o predecir los lugares a los que viajará un determinado perfil. Al aumentar la cantidad de información sobre la que se trabaja, esta mejora puede resultar bastante útil, ya que aumenta la cantidad de información bien valorada que podrían recibir los usuarios.

Referencias

[1] Organización Mundial del Turismo, *"¿Por qué el Turismo?"*, [Online]. Available: https://www2.unwto.org/es/content/por-que-el-turismo [Accessed: 20/09/2019]

[2] UNWTO, UNWTO, WTO. UNWTO annual report 2017. 2017.

[3] Daniel Cruz López de Ochoa, *"Retos del turismo en España durante los próximos años", [Online]. Available:* https://repositorio.comillas.edu/jspui/bitstream/11531/3775/1/TFG000642.pdf [Accessed: 20/09/2019]

[4] Picornell, C. (2015). Los impactos del turismo. *Papers de turisme*, vol. 11, 65-91.

[5] HosteleriaDigital, *"Beneficios económicos y fiscalidad del turismo en España"*, [Online]. Available: https://www.hosteleriadigital.es/2019/01/16/beneficios-economicos-y-fiscalidad-del-turismo-en-espana/ [Accessed: 20/09/2019]

[6] Galicia en Pie, *"Vas a visitor la Playa de las Catedrales? Necesitarás reservar una entrada",* [Online]. Available: http://www.galiciaenpie.com/blog/vas-a-visitar-la-playa-de-las-catedrales-necesitaras-reservar-una-entrada/ [Accessed: 17/06/2019]

[7] UNWTO, *"Tourism Highlights",* [Online]. Available: http://mkt.unwto.org/sites/all/files/docpdf/unwtohighlights11enlr_3.pdf [Accessed: 20/09/2019]

[8] Ismael Nafría, *"España sigue batiendo récords de turistas internacionales: ranking por países y CC.AA."*, [Online]. Available: https://www.lavanguardia.com/vangdata/20150922/54435412973/espana-sigue-batiendo-records-de-turistas-internacionales-ranking-por-paises-y-cc-aa.html [Accessed: 20/09/2019]

[9] Instituto Nacional de Estadística, *"Encuesta de Gasto Turístico"*, [Online]. Available: http://www.ine.es/daco/daco42/egatur/egatur1216.pdf [Accessed: 20/09/2019]

[10] Hosteltur, *"El turismo mundial crece un 6% hasta alcanzar los 1.400 millones de llegadas",* [Online]. Available: https://www.hosteltur.com/126327_el-turismo-mundial-crece-un-6-hasta-alcanzar-1400-millones-de-llegadas.html [Accessed: 20/09/2019]

[11] Shapiro, Stuart C. "*Encyclopedia of artificial intelligence second edition"*. John, 1992.

[12] Y. Dodge, ed., "*The Concise Encyclopedia of Statistics"*. Springer New York, 2008

[13] T. J. Cleophas and A. H. Zwinderman, eds., "*Machine Learning in Medicine"*. Springer Netherlands, p. 166, 2013

[14] C. Sammut and G. I. Webb, eds., *Encyclopedia of Machine Learning*. Springer, p.549-550 2010

[15] D. Y. Singh and A. S. Chauhan, "Neural networks in data mining," *Journal of Theoretical and Applied Information Technology*, vol. 5, no. 1, pp. 37–42, 2009

[16] R. J. Erb, "Introduction to backpropagation neural network computation," *Pharmaceutical Research*, vol. 10, no. 2, pp. 165–170, 1993.

[17] Buscema, M., Back propagation neural networks. *Substance use & misuse*, 1998, vol. 33, no 2, p. 233-270.

[18] L. Liu and M. T. Özsu, eds., *Encyclopedia of Database Systems*. Springer US, 2009

[19] Rokach, Lior, and Oded Z. Maimon. *Data mining with decision trees: theory and applications*. Vol. 69. World scientific, 2008.

[20] Colwell, Will. First Venice and Barcelona: now anti-tourism marches spread across Europe. *The Guardian*, 2017, vol. 10, 2017.

[21] Bergstra, J. S., Bardenet, R., Bengio, Y., & Kégl, B. (2011). Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems* (pp. 2546-2554).

[22] J. V. Tu, "Advantages and disadvantages of using artificial neural networks versus logistic regression for predicting medical outcomes," Journal of Clinical Epidemiology, vol. 49, no. 11, pp. 1225 – 1231, 1996.

[23] Instituto Nacional de Estadística, *"Encuesta de turismo de residentes"*, [Online]. Available: https://www.ine.es/dyngs/INEbase/es/operacion.htm?c=Estadistica_C&cid=1254736176990&menu=resultados&secc=1254736195369&idp=1254735576863 [Accessed: 20/09/2019]

[24] D. M. W. Powers, "Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation," Tech. Rep. SIE-07-001, School of Informatics and Engineering, Flinders University, Adelaide, Australia, 2007.

Glosario

WEKA	Plataforma de software escrita en Java para el aprendizaje automático y la minería de datos.
SQL	Structured Query Language: Lenguaje de dominio específico utilizado para administrar y recuperar información de sistemas de gestión de bases de datos relacionales.
INE	Insituto Nacional de Estadística
FRONTUR	Encuesta de movimientos turísticos en fronteras
CSV	Comma-Separated Values, archivo separado por comas
ARFF	Attribute-Relation File Format, archive utilizado para la herramienta WEKA

Anexos

A Tablas de códigos identificadores para atributos

1. Motivo

Código	Motiv
	Motivos Personales
	Ocio, recreo y vacaciones
1	Turismo de sol y playa
2	Turismo cultural
3	Turismo de naturaleza
4	Turismo gastronómico
5	Turismo deportivo
6	Turismo termal y de bienestar (relax, belleza, desintoxicación, …)
7	Otro tipo de turismo de ocio (asistencia a ferias como particular, fiestas patronales, …)
	Visitas a familiares o amigos
8	Visitas a familiares o amigos (bodas, funerales, reuniones con amigos, …)
	Otros motivos
9	Turismo de compras

Figura B-0-1 Listado de motivos posibles para la realización de un viaje

2. Tipo de viaje

Código	Tipoviaj
1	Viajes de puente
2	Viajes de fin de semana
3	Trabajo
4	Estudio
5	Desplazamiento al centro de estudio
6	Desplazamiento al centro de trabajo
7	Vacaciones de verano
8	Vacaciones de Navidad
9	Vacaciones de Semana Santa
10	Otros viajes

Figura B-0-2 Listado de tipos de viajes posibles

3. Tipo de alojamiento

Código	Alojaprin
	De mercado
1	Hotel o apartahotel
2	Pensión, hostal, motel, fonda, casa de huéspedes
3	Vivienda completa en alquiler (incluye apartamentos turísticos)
4	Habitación en alquiler en vivienda particular
5	Alojamiento turismo rural
6	Albergue
7	Camping
8	Crucero
9	Otros alojamientos de mercado
	No de mercado
10	Vivienda en propiedad
11	Viviendas de familiares, amigos o empresa cedidas gratuitamente
12	Viviendas de uso compartido (multipropiedad)
13	Viviendas intercambiadas
14	Otros alojamientos no de mercado

Figura B-0-3 Listado de posibles tipos de alojamiento del turista

4. Principal medio de transporte

Código	Transprin
1	**Transporte aéreo**
	Transporte Marítimo o Fluvial
2	Crucero
3	Ferry
4	Embarcación náutica propia, cedida o alquilada
	Transporte terrestre
5	Automóvil u otros vehículos particulares propios o cedidos
6	Automóvil u otros vehículos particulares alquilados sin conductor a empresas de alquiler
7	Taxis u otros vehículos particulares alquilados con conductor a empresas de alquiler o transporte
8	Automóvil u otros vehículos compartidos con pago al conductor
9	Autubús
10	Tren
11	Transporte terrestre no motorizado (bici, andando, a caballo, …)
12	Otro medio de transporte

Figura B-0-4 Listado de medios de transporte posibles

5. Comunidades autónomas

Código	Nombre
0	Extranjero
1	Andalucía
2	Aragón
3	Asturias, Principado de
4	Balears, Illes
5	Canarias
6	Cantabria
7	Castilla y León
8	Castilla - La Mancha
9	Cataluña
10	Comunitat Valenciana
11	Extremadura
12	Galicia
13	Madrid, Comunidad de
14	Murcia, Región de
15	Navarra, Comunidad Foral de
16	País Vasco
17	Rioja, La
18	Ceuta
19	Melilla

Figura B-0-5 Listado de comunidades autónomas

6. Provincias

Código Provincia	Provincia	Código Provincia	Provincia
0	Extranjero	26	Rioja (La)
1	Álava	27	Lugo
2	Albacete	28	Madrid
3	Alicante/Alacant	29	Málaga
4	Almería	30	Murcia
5	Ávila	31	Navarra
6	Badajoz	32	Ourense
7	Balears (Illes)	33	Asturias
8	Barcelona	34	Palencia
9	Burgos	35	Palmas (Las)
10	Cáceres	36	Pontevedra
11	Cádiz	37	Salamanca
12	Castellón/Castelló	38	Santa Cruz de Tenerife
13	Ciudad Real	39	Cantabria
14	Córdoba	40	Segovia
15	Coruña (A)	41	Sevilla
16	Cuenca	42	Soria
17	Girona	43	Tarragona
18	Granada	44	Teruel

19	Guadalajara	45	Toledo
20	Guipúzcoa	46	Valencia/València
21	Huelva	47	Valladolid
22	Huesca	48	Vizcaya
23	Jaén	49	Zamora
24	León	50	Zaragoza
25	Lleida	51	Ceuta
		52	Melilla

Figura B-0-6 Listado de provincias

7. Tamaño municipio

Código	Tamaño municipio
1	100.000 o más habitantes
2	De 50.000 a 99.999 habitantes
3	De 20.000 a 49.999 habitantes
4	De 10.000 a 19.999 habitantes
5	Menos de 10.000 habitantes

Figura B-0-7 Listado de posibles tamaños de municipio

8. Grado urbanización

Código	Urba
1	Zona muy poblada
2	Zona media
3	Zona escasamente poblada

Figura B-0-8 Listado de posibles grados de urbanización de la zona

9. Estado civil

Código	Ecivil
1	Soltero/a
2	Casado/a
3	Viudo/a
4	Separado/a
5	Divorciado/a

Figura B-0-9 Listado de posibles estados civiles del turista

10.Tipo de convivencia

Código	Conv
1	Conviviendo con su cónyuge
2	Conviviendo con una pareja de hecho
3	No conviviendo en pareja

Figura B-0-10 Listado de tipos posibles de convivencia

11.Nivel de estudios

Código	Nivelest
1	Educación primaria o inferior
2	Educación secundaria, primera etapa
3	Educación secundaria, segunda etapa
4	Educación superior

Figura B-0-11 Listado de niveles de estudios

12. Actividad económica

Código	Relaecon
1	Ocupado
2	Parado
3	Jubilado
4	Resto inactivos

Figura B-0-12 Listado de posibles actividades económicas

13. Situación profesional

Código	Sitprof
1	Empresario, profesional o trabajador por cuenta propia que emplea a otras personas
2	Empresario, profesional o trabajador por cuenta propia que no emplea a otras personas
3	Asalariado o trabajador por cuenta ajena con contrato indefinido
4	Asalariado o trabajador por cuenta ajena con contrato eventual o temporal

Figura B-0-13 Listado de posibles situaciones profesionales

14.Tipo de hogar

Código	Tiphogar
1	Hogar unipersonal
2	Padre o madre sólo que convive con algún hijo
3	Pareja sin hijos que conviven en el hogar
4	Pareja con hijos que conviven en el hogar
5	Otro tipo de hogar

Figura B-0-14 Listado de tipos de hogar posibles

15.Ingresos en el hogar

Código	Ingr_Hog
1	Hasta 999 euros
2	De 1000 a 1499 euros
3	De 1500 a 2499 euros
4	De 2500 a 3499 euros
5	De 3500 a 4999 euros
6	5000 euros o más
9	No contesta

Figura B-0-15 Listado de posibles rangos de ingresos en el hogar

B Gráficas de resultados adicionales

1. Clase Fidelidad

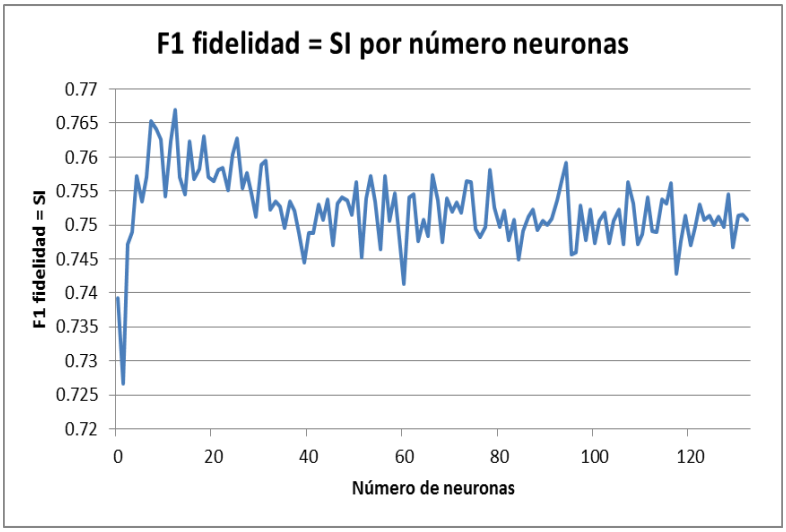

Figura B-0-16 Relación entre F1-measure *fidelidad = SI* y número de neuronas

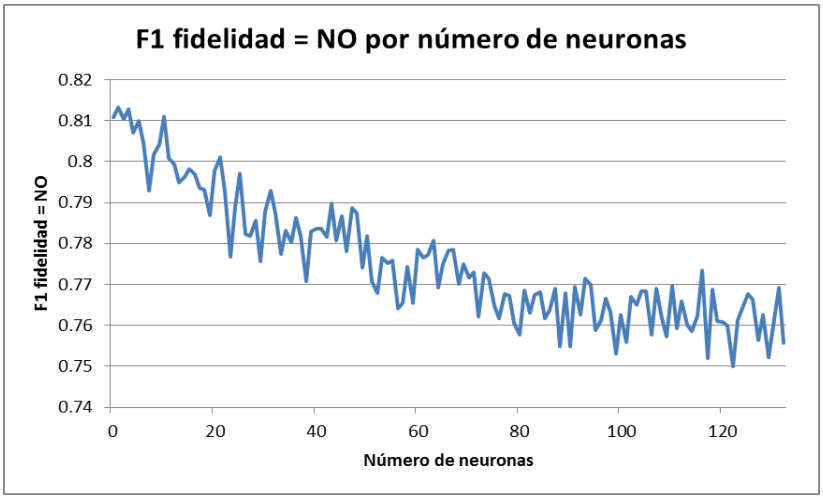

Figura B-0-17 Relación entre F1-measure *fidelidad = NO* y número de neuronas

2. Clase Satisfacción

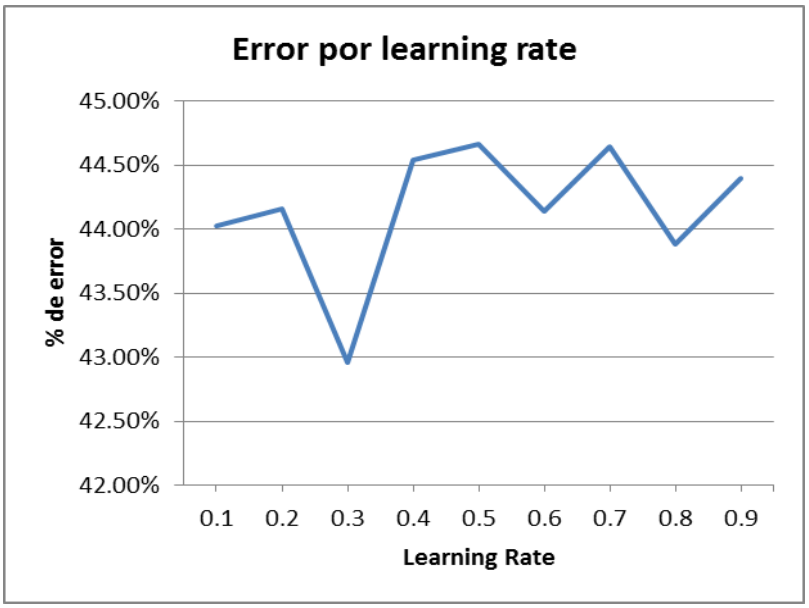

Figura B-0-18 Relación entre el % de error y learning rate

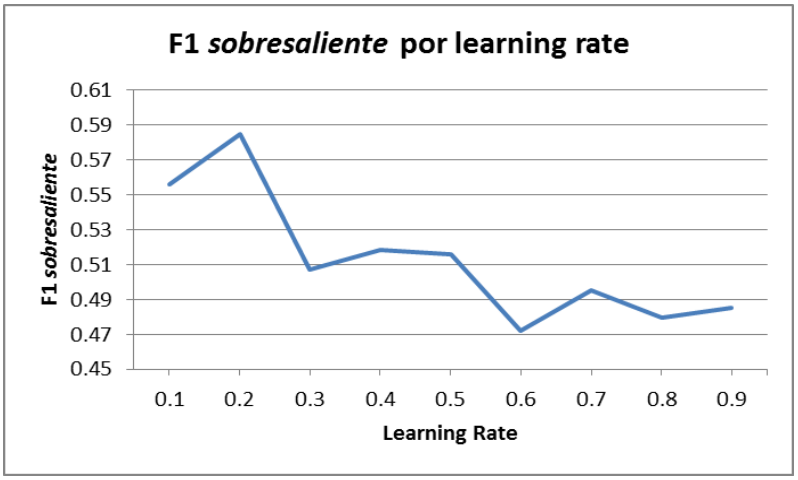

Figura B-0-19 Relación entre F1-measure *sobresaliente* y learning rate

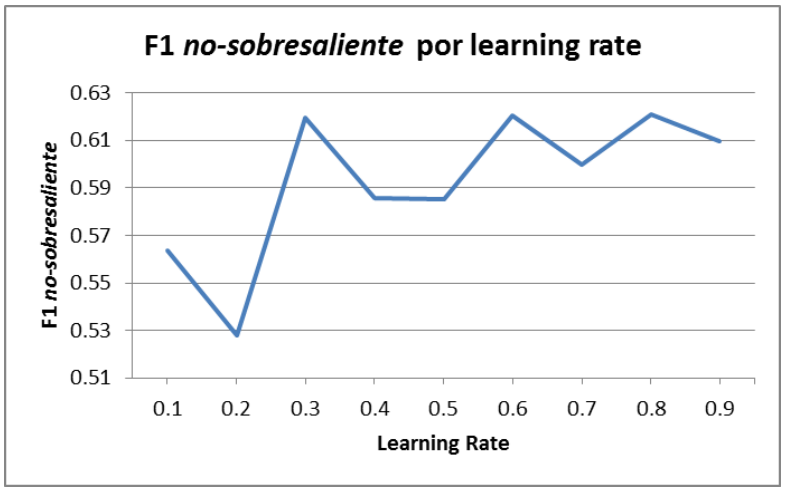

Figura B-0-20 Relación entre F1-measure *no-sobresaliente* y learning rate

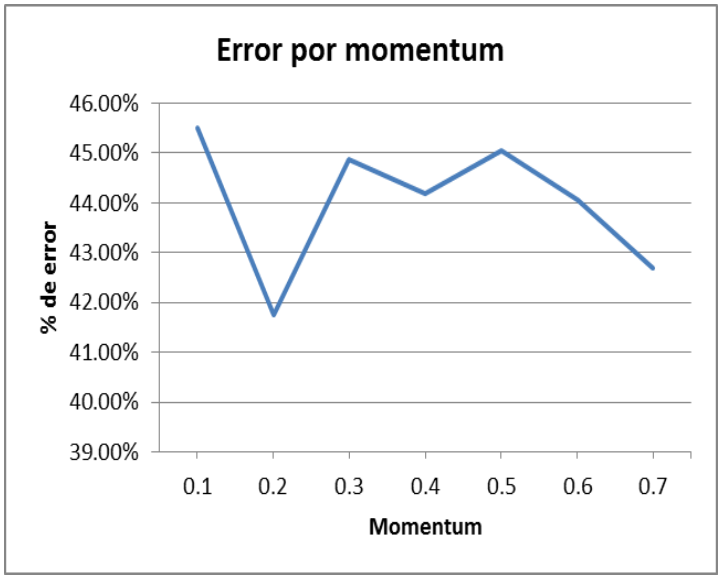

Figura B-0-21 Relación entre el % de error y momentum

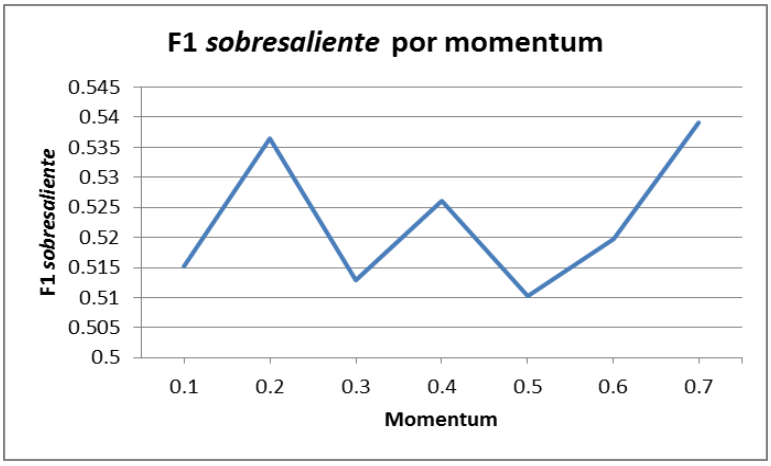

Figura B-0-22 Relación entre F1-measure *sobresaliente* y momentum

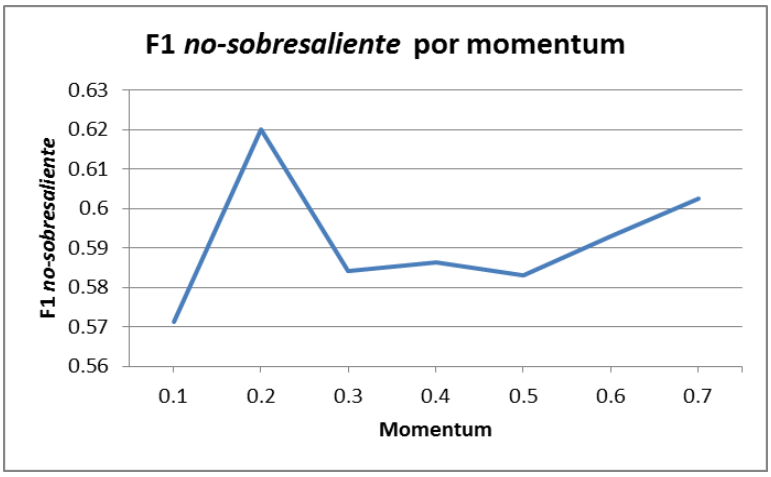

Figura B-0-23 Relación entre F1-measure *no-sobresaliente* y momentum